



TUGAS AKHIR - TM141585

**DESAIN DAN ANALISA PENENTUAN PARAMETER
GAIN SISTEM PENGENDALI *PROPORTIONAL-
INTEGRAL* DENGAN METODE *NEURAL NETWORK*
PADA SISTEM *TURRET GUN***

**DERIS TRIANA NOOR
NRP. 2112100115**

**DOSEN PEMBIMBING
ARIF WAHJUDI, S.T., M.T., Ph.D**

**JURUSAN TEKNIK MESIN
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2017**



FINAL PROJECT - TM141585

**DESIGN AND ANALYSIS PARAMETER GAIN NEURAL-
NETWORK-BASED PROPORTIONAL-INTEGRAL
CONTROLLER FOR TURRET-GUN SYSTEM**

**DERIS TRIANA NOOR
NRP. 2112100115**

**ADVISOR
ARIF WAHJUDI, S.T., M.T., Ph.D**

**MECHANICAL ENGINEERING DEPARTMENT
Faculty Of Industrial Technology
Sepuluh Nopember Institute Of Technology
Surabaya 2017**

**DESAIN DAN ANALISA PENENTUAN PARAMETER
GAIN SISTEM PENGENDALI *PROPORTIONAL-
INTEGRAL* DENGAN METODE *NEURAL NETWORK*
PADA SISTEM *TURRET GUN***

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Teknik
pada
Program Studi S-1 Jurusan Teknik Mesin
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember

Oleh :

DERIS TRIANA NOOR

NRP. 2112 100 115

Disetujui oleh Tim Penguji Tugas Akhir :

1. Arif Wahjudi, ST, MT, PhD. (Pembimbing)
NIP. 197303222001121001
2. Prof. Dr. Ing. I Made Londen B., ME. (Penguji I)
NIP. 195811061986011002
3. Hendro Nurhadi, Dipl. Ing., PhD. (Penguji II)
NIP. 197511202002121002
4. Dinny Harnany, ST, MSc. (Penguji III)
NIP. 2100201405001

SURABAYA

JANUARI, 2017

[Halaman ini sengaja dikosongkan]

DESAIN DAN ANALISA PENENTUAN PARAMETER GAIN SISTEM PENGENDALI *PROPORTIONAL- INTEGRAL* DENGAN METODE *NEURAL NETWORK* PADA SISTEM *TURRET GUN*

Nama : Deris Triana Noor
NRP : 2112100115
Jurusan : TEKNIK MESIN FTI-ITS
Dosen Pembimbing : ARIF WAHJUDI, ST., MT., PhD.

ABSTRAK

Parameter dalam sistem pengendali PID untuk sistem turret gun ditentukan dengan cara menurunkan persamaan diferensial sistem pengendali tersebut. Penurunan persamaan diferensial memakan banyak waktu dan sulit untuk dilakukan. Terutama ketika sistem yang dikontrol sangat rumit dan sulit untuk disederhanakan, sehingga sistem tidak dapat diwakili oleh suatu persamaan. Untuk menanggulangi masalah tersebut, digunakanlah neural network untuk menentukan parameter sistem pengendali. Sistem pengendali yang digunakan adalah sistem pengendali proportional-integral yang hanya memerlukan nilai proportional gain (K_P) dan integral gain (K_I). Nilai K_P dan K_I digunakan sebagai faktor pengali pada nilai error yang menjadi input dari sistem pengendali proportional-integral dan input neural network. Dengan demikian, nilai dari K_P dan K_I pada sistem pengendali proportional-integral diperoleh melalui pelatihan neural network.

Dari Tugas Akhir ini didapatkan hasil nilai K_P dan K_I yang dapat berubah setiap waktu sesuai error yang terjadi pada sistem. Pada uji error 10 sudut yang dipilih secara acak, didapatkan bahwa nilai sudut aktual dari sudut azimuth dan sudut elevasi sudah mendekati nilai inputnya dengan nilai root mean square error (RMSE) adalah 0.0393063 untuk sudut azimuth dan 1.1621708 untuk sudut elevasi. Respon sistem turret gun dengan Neural Network Proportional-Integral (NN-PI) controller dapat menurunkan %overshoot 63,493%, settling time 19,183%, dan

steady-state error 27,064%. Sementara untuk sudut elevasi, Neural Network Proportional-Integral controller hanya dapat menurunkan steady-state error yaitu 34,168 %. Sedangkan untuk settling time mengalami kenaikan 16,463% dan %overshoot juga mengalami kenaikan 75,465%.

Kata kunci: turret gun, neural network, proportional-integral control, azimuth, elevasi.

DESIGN AND ANALYSIS PARAMETER GAIN NEURAL-NETWORK-BASED PROPORTIONAL-INTEGRAL CONTROLLER FOR TURRET-GUN SYSTEM

Name : Deris Triana Noor
NRP : 2112100115
Department : TEKNIK MESIN FTI-ITS
Advisor : ARIF WAHJUDI, ST., MT., PhD.

ABSTRACT

Parameters in PID controller for turret-gun system are determined by derivation of the differential function. The derivation might be difficult and take many times to be solved, especially if the systems are complex and difficult to obtain the exact function which has the same behavior as the system. To solve the problem, neural network are proposed to determine the parameters of the controller. The controller which applied in this project was Proportional-Integral (PI) controller. It has two parameters which was proportional gain (K_P) and integral gain (K_I). Both were used as gain factor for the error value which at the same time have a role as an input value to the controller and the neural network. So, the gain value for the PI controller determined by the neural network training according to the error value.

The result of this final project was obtain the parameter gain value that has the ability to adapt in every condition according to error value that happens in the system. From the error value test, ten set points were choosed randomly. The result of this test was the azimuth angle and elevation angle have the ability to keep up with the set point. Despite the ability of the controller, there was still an error in every angle. The root mean square error (RMSE) was 0.0393063 for the azimuth angle and 1.1621708 for elevation angle. Response of the system with neural network proportional-integral (NN-PI) controller for azimuth angle have the ability to decrease the %overshoot value to 63,493 %, settling time value to 19,183 %, and the steady state error value to 27,064 %. While for

the elevation angle, the controller only have the ability to decrease the steady state error value to 34,168 %. But for settling time value has increased to 16,463 % and %overshoot value to 75,465 %.

Keyword: azimuth, elevation, neural network, proportional-integral, turret-gun

KATA PENGANTAR

Puji dan syukur marilah kita panjatkan kehadiran Tuhan Yang Maha Esa. Karena setiap hidayah dan karunia-Nya, penulis mampu menyelesaikan Tugas Akhir yang berjudul **Desain dan Analisa Penentuan Parameter *Gain* Sistem Pengendali *Proportional-Integral* dengan Metode *Neural Network* pada Sistem *Turret-Gun*.**

Tidak lupa ucapan terima kasih penulis sampaikan kepada berbagai pihak yang telah membantu dalam penyelesaian Tugas Akhir ini. Tanpa orang-orang tersebut, mustahil laporan tugas akhir ini dapat diselesaikan dengan baik. Adapun ucapan terima kasih penulis sampaikan kepada:

1. **Kedua orang tua** penulis yang selalu mendoakan, mendidik, dan memberi semangat putra-putrinya untuk melakukan yang terbaik.
2. **Bapak Arif Wahjudi, S.T., M.T., Ph.D.,** selaku dosen pembimbing Tugas Akhir yang selalu memberikan ilmu, kritik dan saran, serta bimbingannya selama proses penyelesaian Tugas Akhir ini.
3. **Bapak Prof. Dr. Ing. I Made Londen Batan, M.E., Bapak Hendro Nurhadi, Dipl. Ing., Ph.D., dan Ibu Dinny Harnany, S.T., M.T.,** selaku dosen penguji Tugas Akhir, yang telah menyempatkan berbagi ilmu, sehingga tugas akhir ini dapat terselesaikan dengan baik.
4. **Bapak Prof. Dr. Ing. Suhardjono, M.Sc.,** selaku dosen wali yang telah membimbing penulis selama masa studi di Teknik Mesin ITS.
5. **Agung Kartika Fibrianto dan Febriana Pusparaniayu Pasa** selaku partner dalam pembuatan Tugas Akhir ini.
6. Segenap keluarga besar Laboratorium Perancangan dan Pengembangan Produk yang selalu memberikan semangat dan motivasi kepada penulis.

7. Dosen dan karyawan Jurusan Teknik Mesin FTI-ITS yang telah memberikan ilmu, pengalaman, dan bantuannya selama masa studi.
8. Serta semua pihak yang secara langsung ataupun tidak langsung terlibat dalam proses penyelesaian tugas akhir ini.

Dengan selesainya Tugas Akhir ini, penulis berharap laporan ini dapat bermanfaat khususnya bagi penulis sendiri dan umumnya untuk kita semua.

Surabaya, 20 Januari 2017

Deris Triana Noor

DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PENGESAHAN	Error! Bookmark not defined.
ABSTRAK.....	v
ABSTRACT	vii
KATA PENGANTAR	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR.....	xiii
DAFTAR TABEL.....	xv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan.....	2
1.4 Batasan Masalah.....	2
1.5 Manfaat Tugas Akhir.....	3
BAB II TINJAUAN PUSTAKA.....	5
2.1 Tinjauan pustaka.....	5
2.2 Model <i>Turret Gun</i>	6
2.3 <i>Proportional Control</i>	8
2.4 <i>Integral Control</i>	9
2.5 <i>Neural network</i>	11
2.6 Respon sistem.....	13
2.6.1 Respon Peralihan (<i>transient response</i>).....	13
2.6.2 Karakteristik Respon Waktu	14
2.6.3 Spesifikasi Respon <i>Transient</i>	15
2.6.4 Karakteristik Respon Waktu (<i>Time Response</i>) ...	16
BAB III METODE PENELITIAN	17
3.1 Diagram Alir Penelitian.....	17
3.2 Langkah-langkah Penelitian	19
BAB IV PENENTUAN PARAMETER GAIN DENGAN NEURAL NETWORK.....	23
4.1 Pengambilan Data.....	23
4.2 Hasil Pengambilan Data	24
4.3 Pelatihan Neural Network	26

4.2.1	Pelatihan <i>Neural Network</i> untuk Sudut Azimuth ..	27
4.2.2	Pelatihan <i>Neural Network</i> untuk Sudut Elevasi	31
4.4	Blok Diagram dan Model Simulink dari Sistem <i>Turret-Gun</i> serta Sistem Pengendalinya	36
BAB V SIMULASI SISTEM DENGAN PROPORTIONAL-INTEGRAL CONTROLLER		43
5.1	Simulasi	43
5.2	Perbandingan PID Controller dengan Neural Network PI Controller.....	49
5.3	Simulasi Sistem dengan Variasi Sudut	53
5.4	Parameter Gain Proportional-Integral (PI) pada <i>Controller</i> 58	
BAB VI KESIMPULAN DAN SARAN		65
6.1	Kesimpulan.....	65
6.2	Saran	65
DAFTAR PUSTAKA		67
LAMPIRAN		69

DAFTAR GAMBAR

Gambar 2.1 Sistem turret dan gun [2]	6
Gambar 2.2 diagram control proporsional [3]	8
Gambar 2.3 Diagram control proporsional motor DC [3]	8
Gambar 2.4 Respon output controller P [3]	9
Gambar 2.5 Diagram kontrol PI untuk sebuah motor DC [3]	10
Gambar 2.6 Respon kontroler PI [3]	10
Gambar 2.7 Node dari neuron [4]	11
Gambar 2.8 Struktur dari neural network [5]	12
Gambar 2.9 Sinyal transient response [7]	13
Gambar 2.10 Spesifikasi respon transient [7]	15
Gambar 3.1 Diagram alir Penelitian	18
Gambar 3.2 Diagram alir untuk membangun model Neural network	19
Gambar 4.1 Blok diagram sistem <i>turret-gun</i> pada penelitian sebelumnya [11]	23
Gambar 4.2 Toolbox Data Manager	26
Gambar 4.3 Toolbox Create Network or Data	27
Gambar 4.4 Performa dari neural network sudut azimuth setelah pelatihan	29
Gambar 4.5 Blok Neural network Proportional-Integral (NN-PI) controller untuk sudut azimuth	30
Gambar 4.6 Performa dari neural network untuk sudut elevasi setelah pelatihan	33
Gambar 4.7 Blok Neural network Proportional-Integral (NN-PI) controller untuk sudut elevasi	34
Gambar 4.8 Blok diagram sistem <i>turret-gun dengan Neural Network Proportional-Integral controller</i> . Error! Bookmark not defined.	
Gambar 4.9 Blok Simulink untuk sistem turret gun dengan NN-PI controller	Error! Bookmark not defined.
Gambar 4.10 Blok neural network pada model Simulink	39
Gambar 4.11 Blok Simulink untuk sistem pengendali turret gun	40

Gambar 4.12 Model Simulink untuk sistem turret gun **Error! Bookmark not defined.**

Gambar 5.1 Sistem Turret-Gun setelah diberikan sistem pengendali..... **Error! Bookmark not defined.**

Gambar 5.2 Respon sistem dengan sudut azimuth 20 derajat dan sudut elevasi 60 derajat **Error! Bookmark not defined.**

Gambar 5.3 Respon sistem dengan sudut azimuth 10 derajat dan sudut elevasi 40 derajat **Error! Bookmark not defined.**

Gambar 5.4 Respon sistem dengan sudut azimuth 40 derajat dan sudut elevasi 20 derajat **Error! Bookmark not defined.**

Gambar 5.5 Grafik respon sistem turret-gun dengan variasi set point sudut azimuth (0, 10, 20, 60) dan sudut elevasi (0, 40, 30, 10)..... **Error! Bookmark not defined.**

Gambar 5.6 Grafik respon sistem turret-gun dengan variasi set point sudut azimuth (60, 30, 10, 40) dan sudut elevasi (20, 60, 50, 10)..... **Error! Bookmark not defined.**

Gambar 5.7 Grafik respon sistem turret-gun dengan variasi set point sudut azimuth (40, 60, 10, 40) dan sudut elevasi (10, 40, 20, 60)..... **Error! Bookmark not defined.**

Gambar 5.8 Grafik nilai K_p Azimuth pada sudut azimuth 20 derajat **Error! Bookmark not defined.**

Gambar 5.9 Grafik nilai K_i azimuth pada sudut azimuth 20 derajat **Error! Bookmark not defined.**

Gambar 5.10 Grafik nilai K_p elevasi pada sudut elevasi 60 derajat **Error! Bookmark not defined.**

Gambar 5.11 Grafik nilai K_i elevasi pada sudut elevasi 60 derajat **Error! Bookmark not defined.**

DAFTAR TABEL

Tabel 4.1 Data untuk pelatihan <i>neural network</i> untuk sudut azimuth	25
Tabel 4.2 Data untuk pelatihan <i>neural network</i> untuk sudut elevasi.....	25
Tabel 4.3 Hasil <i>trial</i> dan <i>error Neural Network</i> sudut azimuth ..	28
Tabel 4.4 Hasil <i>trial</i> dan <i>error Neural Network</i> untuk sudut elevasi.....	31
Tabel 5.1 Error posisi rata-rata sistem turret gun pada simulasi selama 15 detik.....	49
Tabel 5.2 Performa sistem <i>turret-gun</i> dengan <i>PID controller</i> untuk sudut azimuth	50
Tabel 5.3 Performa sistem <i>turret-gun</i> dengan <i>PID controller</i> untuk sudut elevasi	51
Tabel 5.4 Performa sistem <i>turret-gun</i> dengan <i>NN-PI controller</i> untuk sudut azimuth	52
Tabel 5.5 Performa sistem <i>turret-gun</i> dengan <i>NN-PI controller</i> untuk sudut elevasi	53
Tabel 1 Hasil simulasi sistem dengan set poin -60 sampai 60 derajat untuk sudut azimuth	69
Tabel 2 Hasil simulasi sistem dengan set poin -20 sampai 60 derajat untuk sudut elevasi	70

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

1.1 Latar Belakang

Saat ini, penelitian tentang senjata otomatis telah berkembang sangat pesat di dunia. Hal ini dikarenakan kegunaan senjata otomatis yang sangat penting dalam badan pertahanan suatu Negara. Selayaknya senjata otomatis haruslah mempunyai fungsi dalam pengenalan target. Namun masih sulit menciptakan suatu teknologi senjata otomatis yang dapat menentukan targetnya sendiri dengan penggunaan pengendalian otomatis.

PID *controller* sejauh ini adalah *controller* yang paling banyak digunakan di industri. Dalam pengendalian PID, terdapat parameter-parameter yang harus ditentukan. Terdapat tiga parameter yaitu nilai *proportional gain* (K_P), *integral gain* (K_I), dan *derivative gain* (K_D). Bagian *proportional* menunjukkan nilai error, *integral* menunjukkan nilai rata-rata dari error, dan *derivative* adalah nilai prediksi dari error dengan pendekatan *linier extrapolation* [10]. Pengaruh nilai K_P , K_I , dan K_D pada *transient response* adalah untuk mempercepat respon, mengurangi *steady-state error*, dan memperkecil *overshoot*. Jika sistem berperilaku dinamis, maka nilai dari parameter-parameter tersebut diperoleh dari persamaan differensial. Namun, terdapat beberapa kesulitan jika menggunakan persamaan differensial. Salah satunya adalah sistem yang cukup kompleks sehingga sulit untuk menemukan persamaan yang dapat mewakili sistem yang ingin dikendalikan. Ketika persamaan dari sistem tidak dapat mewakili sistem secara keseluruhan, maka hasil yang diperoleh tidak dapat digunakan. Selain itu, butuh waktu yang cukup lama untuk menurunkan persamaan dari sistem.

Untuk mencari solusi dari permasalahan diatas adalah dengan menggunakan salah satu metode *intelligent* yaitu *neural network*. *Neural network* adalah gabungan koneksi-koneksi dari elemen proses (*synapse*), *units* atau *nodes* yang saling terhubung yang didasarkan pada fungsi *neuron* hewan. Kemampuan proses dari

network terdapat pada nilai dari koneksi antar-unit (*weights*) yang diperoleh dari proses adaptasi, atau mempelajari pola dari suatu data input dan output [9]. Output dari proses adaptasi ini adalah untuk mendapatkan persamaan dari *proportional gain* dan *Integral gain* yang digunakan pada sistem pengendali *proportional-integral*.

Untuk mendapatkan persamaan *gain* yang sesuai dengan output yang diinginkan, maka dibutuhkan proses adaptasi dari *neural network*. Oleh karena itu, tugas akhir ini diajukan untuk mengetahui apakah proses adaptasi *neural network* yang digunakan sudah sesuai dengan sistem yang dikendalikan.

1.2 Rumusan Masalah

Rumusan masalah dalam laporan tugas akhir ini adalah sebagai berikut:

1. Bagaimana mengontrol sistem dengan menggunakan *neural network*
2. Bagaimana mendapatkan nilai K_P dan K_I melalui *neural network* pada sistem pengendali
3. Bagaimana meningkatkan respon dari sistem pengendali pada *turret gun*

1.3 Tujuan

Adapun tujuan dari penulisan tugas akhir ini adalah:

1. Mengendalikan torsi dari motor penggerak *turret gun*
2. Mendapatkan nilai K_P dan K_I melalui *neural network*
3. Meningkatkan respon dari sistem pada *turret gun*

1.4 Batasan Masalah

Batasan masalah pada penulisan laporan tugas akhir ini adalah sebagai berikut;

1. Persamaan *control* diperoleh dengan metode *neural network*
2. Rancang bangun alat sistem *turret gun* tidak dibahas
3. Jumlah *layer* ditentukan dengan metode *trial and error*
4. *Weight* dan *bias* ditentukan dengan *software* Matlab

1.5 Manfaat Tugas Akhir

Tugas akhir ini diharapkan dapat memberi manfaat diantaranya sebagai berikut:

1. Memperkaya dan memperdalam wacana dalam bidang ilmu mekatronika khususnya mengenai sistem *auto-tuning controller*
2. Diperoleh pengetahuan sistem kendali *auto-tuning* yang diterapkan pada sistem *turret gun*

[Halaman ini sengaja dikosongkan]

BAB II TINJAUAN PUSTAKA

2.1 Tinjauan pustaka

Penelitian tentang sistem pengendali telah banyak dilakukan dengan berbagai macam metode. Berikut ini adalah beberapa penelitian tentang sistem pengendali. Penelitian [1] sebelumnya menjelaskan desain PID *controller* pada *turret-gun*. *Controller* tersebut diberikan untuk meningkatkan respon dari sistem. *Proportional control* berfungsi untuk memperkuat sinyal kesalahan (sinyal error), sehingga mempercepat keluaran sistem untuk mencapai titik referensi. *Integral control* berfungsi untuk menurunkan *steady-state error*. Sedangkan *derivative control* berfungsi untuk mengurangi efek *overshoot*. Ketiga *control* tersebut digabung menjadi satu kesatuan.

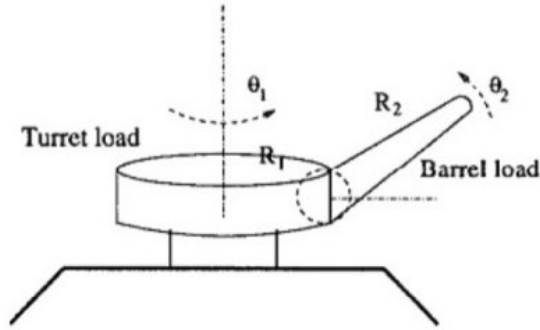
Pada *control* PID, terdapat parameter yang berpengaruh terhadap fungsi *controller*, yaitu nilai K_P , K_I , dan K_D . Nilai tersebut didapat dari *tuning* dengan metode *root locus*. *Root locus* pada suatu sistem secara umum dapat diperbolehkan untuk memilih penguat (*gain*) yang ideal dengan spesifikasi respon *transient* yang diinginkan. Metode tersebut masih menggunakan *tuning* manual dengan mencari nilai K_P , K_I , dan K_D secara perhitungan.

Metode pengendalian senjata pengenalan target yang berbeda juga pernah dilakukan [2]. Mereka meneliti berbagai macam aplikasi metode *control* konvensional pada *automatic turret gun*. Seperti *control* PID (*Proportional–Integral–Derivative controller*), RAC (*Resolved Acceleration Control*) dan RACAFC (*Resolved Acceleration Control Active Force Control*). Sejumlah percobaan telah dilakukan untuk berbagai macam sudut baik di azimuth dan elevasi sumbu *turret gun*. Dari hasil penelitian menunjukkan bahwa untuk sistem ATG dapat bekerja dibawah gangguan, RACAFC menunjukkan performa yang lebih baik dari kedua metode *control*, yaitu metode RAC dan metode PID. Akan tetapi jika percobaan tanpa pemberian beban, hasil yang memuaskan didapat dari RAC

dan RACAFC, sedangkan pada *control* PID tidak. Dikarenakan *control* PID tidak dapat menjangkau semua sudut yang diberikan.

2.2 Model Turret Gun

Model *turret-gun* terdiri dari sistem *turret* dan *gun*. Sistem tersebut diilustrasikan pada gambar 2.1.



Gambar 2.1 Sistem *turret* dan *gun* [2]

Pada sistem tersebut terdapat dua alat yaitu *turret* dan *gun*. *Turret* adalah alat yang pergerakannya memutar sedangkan *gun* adalah alat yang pergerakannya naik turun. Parameter yang dikenakan pada *turret* dan *gun* adalah massa, radius, dan posisi angular. Persamaan dinamikanya sebagai berikut:

$$D(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) = \tau \quad (2.1)$$

Dimana τ adalah torsi, $D(\theta)$ adalah matriks inersia, $C(\theta, \dot{\theta})$ adalah vektor gaya koriolis dan sentrifugal, dan $G(\theta)$ adalah vector gaya gravitasi. Matriks inersia, vektor gaya koriolis dan sentrifugal, serta vektor gaya gravitasi ditunjukkan melalui persamaan berikut:

$$D(\theta) = \begin{bmatrix} D_{11} & 0 \\ 0 & D_{22} \end{bmatrix} \quad (2.2)$$

$$C(\theta, \dot{\theta}) = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \quad (2.3)$$

$$G(\theta) = \begin{bmatrix} 0 & \frac{1}{2} m_2 g R_2 \cos(\theta_2) \end{bmatrix}^T \quad (2.4)$$

Dimana:

$$D_{11} = \frac{1}{2} m_1 R_1^2 + m_2 R_1^2 + m_2 R_1 R_2 \cos(\theta_2) + \frac{1}{3} m_2 R_2^2 \cos^2(\theta_2) \quad (2.5)$$

$$D_{22} = \frac{1}{3} m_2 R_2^2 \quad (2.6)$$

$$C_{11} = (-m_2 R_1 R_2 \sin(\theta_2) \dot{\theta}_2) \quad (2.7)$$

$$C_{12} = \left(\frac{1}{3} m_2 R_2^2 \sin(2\theta_2) \dot{\theta}_1 \right) \quad (2.8)$$

$$C_{21} = \left(\frac{1}{2} m_1 R_1 R_2 \sin(\theta_2) + \frac{1}{6} m_2 R_2^2 \sin(\theta_2) \right) \dot{\theta}_2 \quad (2.9)$$

$$C_{21} = 0 \quad (2.10)$$

Persamaan dinamis pada sistem *turret-gun* merupakan total dari torsi yang ada pada sistem. Oleh karena itu, persamaan dinamis tersebut harus didefinisikan ke dalam sebuah persamaan gerak dengan basis angular untuk mendapatkan percepatan, kecepatan, dan posisi. Persamaan tersebut adalah sebagai berikut:

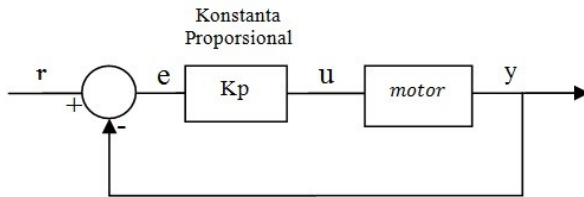
$$\ddot{\theta}_1 = \frac{\tau_1 - C_{11}\dot{\theta}_1 - C_{12}\dot{\theta}_2}{D_{11}} \quad (2.11)$$

$$\ddot{\theta}_2 = \frac{\tau_2 - C_{21}\dot{\theta}_1 - G}{D_{22}} \quad (2.12)$$

Input dari persamaan dinamis tersebut adalah posisi sudut *turret*(θ_1) atau sudut azimuth dan *gun*(θ_2) atau sudut elevasi yang diinginkan serta kecepatan dari *turret* ($\dot{\theta}_1$) dan *gun* ($\dot{\theta}_2$). Sedangkan input *control* dari sistem adalah torsi untuk *turret*(τ_1) dan *gun*(τ_2).

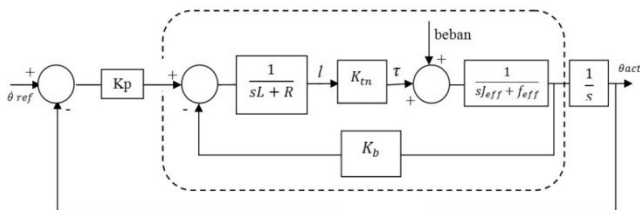
2.3 Proportional Control

Control proporsional berfungsi untuk memperkuat sinyal kesalahan penggerak (sinyal error), sehingga akan mempercepat keluaran sistem mencapai titik referensi. Hubungan antara input *controller* r dengan sinyal error e terlihat pada persamaan $u = K_p \cdot e$.



Gambar 2.2 diagram control proporsional [3]

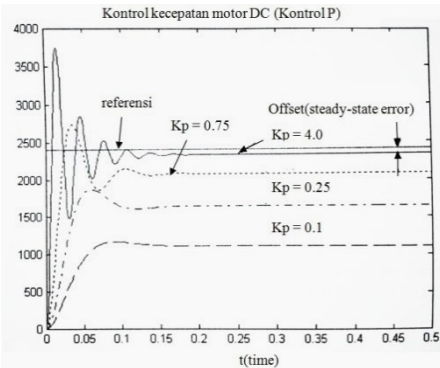
Gambar 2.2 menunjukkan blok diagram yang menggambarkan hubungan antara besaran *setting*, besaran aktual dengan besaran keluaran pengontrol proporsional. Sinyal keasalahan (error) merupakan selisih antara besaran setting dengan besaran aktualnya. Selisih ini akan mempengaruhi pengontrol, untuk mengeluarkan sinyal positif (mempercepat pencapaian harga setting) atau negatif (memperlambat tercapainya harga yang diinginkan).



Gambar 2.3 Diagram control proporsional motor DC [3]

Untuk lebih jelasnya pada contoh kasus *control* kecepatan pada motor DC menggunakan *control* P seperti pada gambar 2.3. Dalam uji simulasi ini, seluruh parameter motor dimasukan ke

dalam blok sistem simulasi sesuai dengan diagram *control* pada gambar diatas. Kecepatan putar referensi yang digunakan adalah 2400 rpm. Skema *control* P di atas diuji dengan memberikan nilai K_P yang berbeda.



Gambar 2.4 Respon output *controller* P [3]

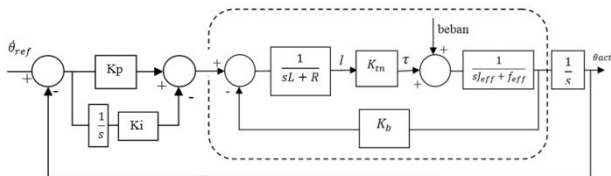
Nampak dalam gambar 2.6 bahwa semakin kecil K_P maka *offset* atau *steady state error* semakin besar. Namun nilai K_P yang terlalu besar akan menyebabkan osilasi pada saat start. *Control* P dapat digunakan sendirian dalam aplikasi. Beberapa aplikasi seperti *control* temperatur pada *heater*, sistem penghematan energi pada sistem *air conditioning* berdasarkan *control* kecepatan motor kompresor sudah cukup memadai menggunakan *control* P saja.

2.4 Integral Control

Fungsi dasar kontrol I adalah menurunkan *steady-state error*. Kontrol I jarang digunakan sendirian dalam aplikasi. Biasanya selalu dikombinasikan dengan kontrol P untuk memperbaiki respon guna mencapai eror minimum. Pengontrol *integral* berfungsi menghasilkan respon sistem yang memiliki kesalahan keadaan stabil nol. Jika sebuah Plant tidak memiliki unsur integrator ($1/s$), pengontrol proposional tidak akan mampu menjamin keluaran sistem dengan kesalahan keadaan stabilnya nol. Dengan pengontrol *integral*, respon sistem dapat diperbaiki, yaitu

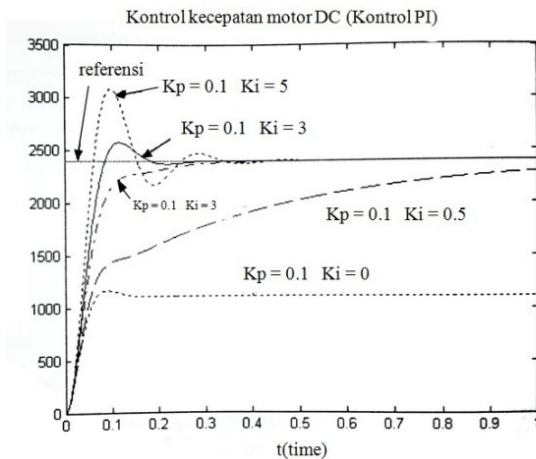
mempunyai kesalahan keadaan stabilnya nol. Pengontrol *Integral* memiliki karakteristik seperti halnya sebuah *integral*.

Keluaran sangat dipengaruhi oleh perubahan yang sebanding dengan nilai sinyal kesalahan. Keluaran pengontrol ini merupakan penjumlahan yang terus menerus dari perubahan masukannya. Kalau sinyal kesalahan tidak mengalami perubahan, keluaran akan menjaga keadaan seperti sebelum terjadinya perubahan masukan. Parameter kontrol I dapat diilustrasikan dalam diagram kontrol motor DC, sebagai berikut



Gambar 2.5 Diagram kontrol PI untuk sebuah motor DC [3]

Diagram simulasi kontroler PI ini diuji pada 2400 rpm dengan K_p di set pada 0.1. Untuk melihat efek penerapan kontrol *integral*, dibandingkan tiga kondisi hasil pemilihan parameter K_I yaitu 0, 0.5, dan 2. $K_I = 0$ menunjukan bahwa kontroler adalah P saja.

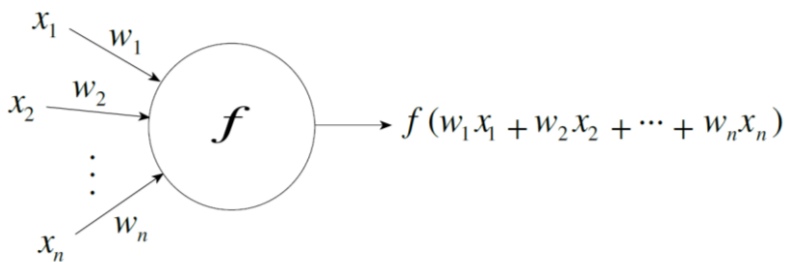


Gambar 2.6 Respon kontroler PI [3]

Dalam Gambar 2.6 diatas nampak bahwa penerapan kontrol P dapat menurunkan *steady-state error*. Namun pemilihan K_I yang terlalu besar dapat menyebabkan sistem berosilasi saat start. Kontrol PI diketahui dipakai secara meluas di dunia industri. Utamanya untuk kasus yang memiliki respon sistem relatif lambat, seperti kontrol temperatur, tekanan fluida, hidrolik dan lain sebagainya.

2.5 Neural network

Neural network adalah salah satu model komputasi yang terdiri dari *neuron-neuron* buatan untuk memproses informasi dari input ke output. *Neuron-neuron* terdiri dari *synapses* dan *nodes*. Kumpulan dari node-node disebut *layer*. *Layer* dapat dibuat lebih dari satu. *Node* digunakan untuk menghimpun fungsi-fungsi.



Gambar 2.7 *Node* dari *neuron* [4]

Gambar 2.5 menunjukkan struktur dari *neuron* dengan n input. Setiap *synapse* dari input dapat mentransmisikan nilai real x . Fungsi f dibangun secara acak didalam *neuron*. Biasanya *synapse* dari input mempunyai nilai *weight*, yang berarti nilai pada input (X_i) dikalikan dengan *weight* (w_i) pada *synapse* tersebut. Nilai-nilai yang ditransmisikan pada *neuron* kemudian dibuat sebuah fungsi f [4].

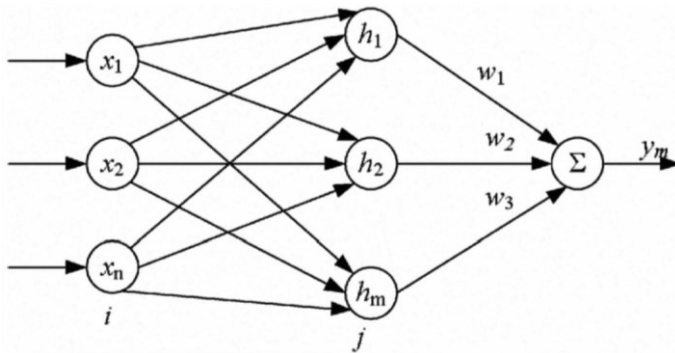
Ada tiga factor yang perlu diperhatikan dalam membuat model *neural network*, yaitu:

1. Struktur dari *node*

2. Topologi dari *network*
3. *Learning algorithm* yang digunakan untuk mendapatkan nilai *weight* [4].

Salah satu penggunaan *neural network* adalah untuk menentukan sebuah fungsi yang mendekati fungsi aslinya. Fungsi asli tersebut *biasanya* tidak diketahui karena sistem terlalu rumit untuk diturunkan ke dalam sebuah persamaan. Penggunaan *neural network* dimaksudkan untuk memperoleh persamaan *gain* dari sistem *control*. *Gain* tersebut akan mempengaruhi sistem sehingga kecepatan respon dari sistem akan meningkat.

Dibawah ini adalah contoh dari *neural network* dengan tiga input dan satu output:



Gambar 2.8 Struktur dari *neural network* [5]

Gambar 2.6 menunjukkan pada *neural network* terdapat tiga *layer*, yaitu *layer input*, *hidden layer* dan *layer output*. Jika sebagai input didefinisikan sebagai $X = [x_1, x_2, \dots, x_n]^T$ dan $H = [h_1, h_2, \dots, h_m]^T$, maka h_j adalah fungsi Gaussian yang didefinisikan sebagai berikut:

$$h_j = \exp\left(-\frac{\|X - c_j\|^2}{2b_j^2}\right), j = 1, 2, \dots, m \quad (2.13)$$

dimana $C_j = [c_{j1}, c_{j2}, \dots, c_{ji}, \dots, c_{jm}]^T$, $B = [b_1, b_2, \dots, b_m]^T$, dan *weight* dari *network* adalah $W = [w_1, w_2, \dots, w_m]^T$, maka output dari *network* tersebut adalah:

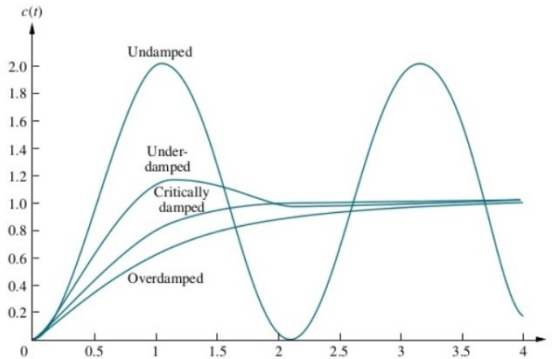
$$y_m(k) = w_1 h_1 + w_2 h_2 + \dots + w_m h_m \quad (2.14)$$

2.6 Respon sistem

Respon sistem atau tanggapan sistem merupakan perubahan perilaku output terhadap perubahan sinyal input. Respon sistem berupa kurva dan dapat menjadi dasar untuk menganalisa karakteristik sistem. Bentuk kurva respon sistem dapat dilihat setelah mendapatkan sinyal input.

2.6.1 Respon Peralihan (*transient response*)

Ketika masukan sebuah sistem mengalami perubahan secara tiba-tiba keluaran sistem membutuhkan waktu untuk merespon perubahan itu. Bentuk peralihan tersebut digambarkan seperti berikut:



Gambar 2.9 Sinyal transient response [7]

Bentuk sinyal respon *transient* dibagi mejadi 4 yaitu:

- *Undamped response*, yaitu respon sistem akan terus berosilasi dengan amplitudo yang semakin membesar, dengan nilai $\delta = 0$

- *Underdamped response*, yaitu output melesat naik untuk mencapai input lalu turun yang kemudian berhenti pada kisaran nilai akhir. Respon ini memiliki efek osilasi dengan nilai $0 < \delta < 1$
- *Overdamped response*, yaitu output tidak melewati nilai input tetapi membutuhkan waktu lama untuk mencapai nilai akhirnya, dengan nilai $\delta = 1$
- *Critically damped response*, respon yang dapat mencapai nilai akhir dengan cepat dan tidak melewati batas input, dengan nilai $\delta > 1$ [7].

Fasa peralihan ini akan berhenti pada kisaran nilai input atau target dimana selisih nilai akhir dengan target disebut *steady state error*. Jika input yang diberikan pada fasa *transient* telah mencapaikondisi *steady state* maka sistem dikatakan stabil. Jika sistem tidak stabil, output mengalami peningkatan terus tanpa batas hingga sistem merusak diri sendiri atau terdapat rangkaian pengaman yang memutus sistem.

2.6.2 Karakteristik Respon Waktu

Merupakan karakteristik respon yang spesifikasi performansinya didasarkan pada pengamatan bentuk respon output sistem terhadap berubahnya waktu. Spesifikasi performansi respon waktu dibagi atas dua tahapan pengamatan, yaitu:

- Spesifikasi respon *transient*

Merupakan spesifikasi respon sistem yang diamati pada saat terjadinya perubahan sinyal input atau gangguan hingga respon memasuki keadaan *steady state*. Parameter untuk mengukur kualitas respon *transient* ini antara lain *rise time*, *delay time*, *peak time*, *settling time*, dan *%overshoot*.

- Spesifikasi respon *steady state*

Merupakan spesifikasi respon sistem yang diamati pada saat respon memasuki keadaan *steady state* hingga waktu tak terbatas. Parameter untuk mengukur kualitas respon *steady state* ini yaitu *%error steady state* [8].

2.6.3 Spesifikasi Respon *Transient*

Terdapat beberapa parameter yang menandakan kualitas respon *transient* antara lain:

- *Rise time (RT)*

Yaitu waktu yang diperlukan kurva untuk naik dari 0.1 ke 0.9 dari respon *steady state*.

- *Settling Time (ST)*

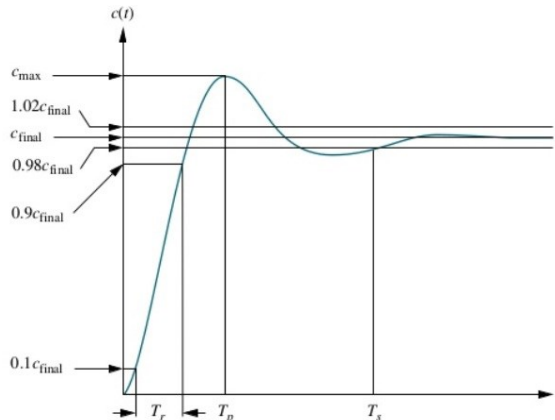
Yaitu waktu yang diperlukan kurva respon untuk mencapai 2% dari nilai *steady state*-nya.

- *Persen Overshoot (OS)*

Yaitu nilai relatif yang menyatakan perbandingan antara nilai tertinggi respon yang melebihi nilai *steady state* dibandingkan dengan nilai *steady state*.

- *Time Peak (PT)*

Yaitu waktu yang diukur mulai $t = 0$ hingga respon pertama kali mencapai puncak maksimum [7].



Gambar 2.10 Spesifikasi respon transient [7]

Risetime, *settling time*, dan *peak time* memberikan informasi mengenai kecepatan dan "kualitas" respon transien. Besaran - besaran ini dapat membantu perancang untuk mencapai

kecepatan yang diinginkan tanpa osilasi atau *overshoot* yang berlebihan.

2.6.4 Karakteristik Respon Waktu (*Time Response*)

Time Response adalah karakteristik respon yang spesifikasi performansinya didasarkan pada pengamatan bentuk respon output sistem terhadap berubahnya waktu [8]. Secara umum spesifikasi performansi respon waktu dapat dibagi atas dua tahapan pengamatan, yaitu;

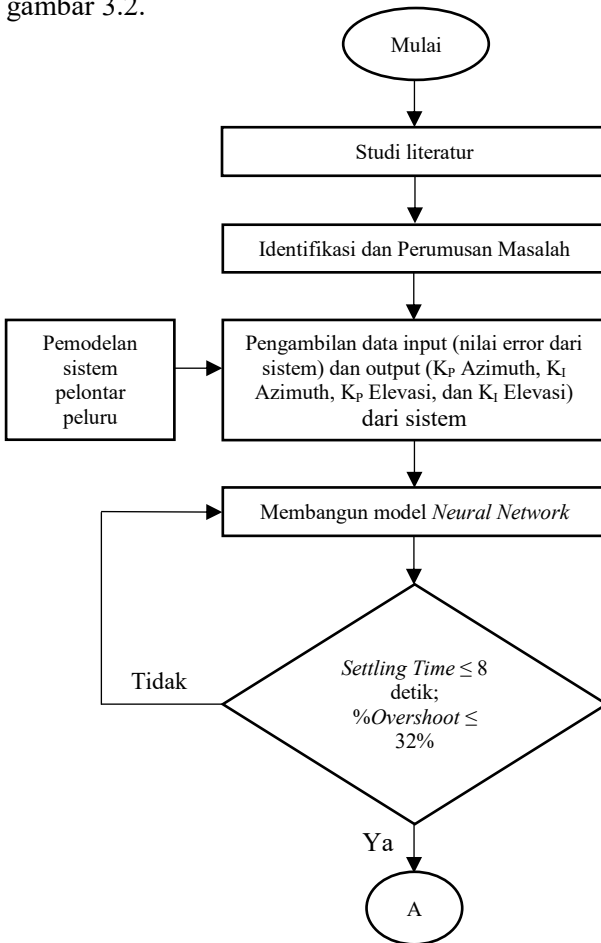
- Spesifikasi Respon *Transient*, adalah spesifikasi respon sistem yang diamati mulai saat terjadinya perubahan sinyal input/ gangguan / beban sampai respon masuk dalam keadaan *steady state*. Tolak ukur yang digunakan untuk mengukur kualitas respon *transient* ini antara lain; *rise time*, *delay time*, *peak time*, *settling time*, dan *%overshoot*.
- Spesifikasi Respon *Steady State*, adalah spesifikasi respon sistem yang diamati mulai saat respon masuk dalam keadaan *steady state* sampai waktu tak terbatas. Tolak ukur yang digunakan untuk mengukur kualitas respon *steady state* ini antara lain; *%error steady state* baik untuk error posisi, error kecepatan maupun error percepatan.

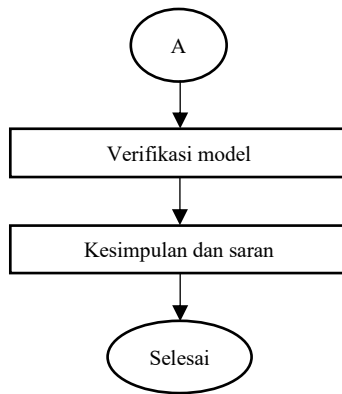
BAB III

METODE PENELITIAN

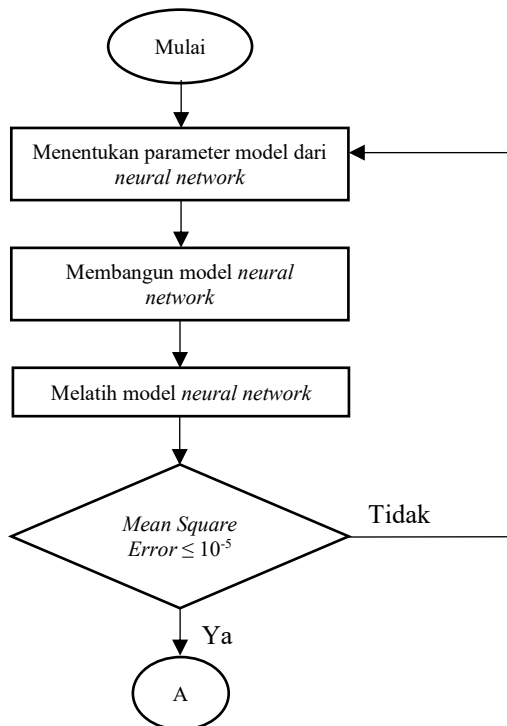
3.1 Diagram Alir Penelitian

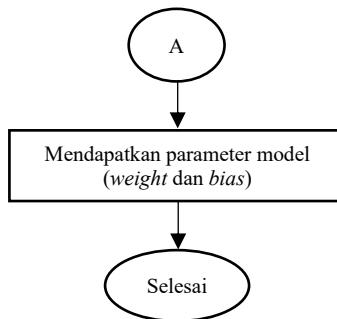
Tahapan penelitian pada tugas akhir ini dilaksanakan sesuai dengan diagram alir pada gambar 3.1 dan penjelasan lebih detail untuk membangun model *Neural Network* dapat dilihat pada gambar 3.2.





Gambar 3.1 Diagram alir Penelitian





Gambar 3.2 Diagram alir untuk membangun model *Neural network*

3.2 Langkah-langkah Penelitian

Langkah-langkah penelitian secara detail pada laporan tugas akhir ini adalah sebagai berikut:

1. Studi literatur

Pada studi literatur ini dilakukan dengan mempelajari literatur-literatur dinataranya buku referensi dan jurnal keilmiahan yang mendukung terhadap penelitian. Studi literature ini difokuskan pada pembelajaran mengenai *neural network* dan sistem pengendali proporsional. Selain itu, mempelajari sistem yang dikontrol juga diperlukan untuk mengetahui karakteristik dari sistem.

2. Perumusan Masalah

Langkah ini adalah menentukan permasalahan yang diangkat pada penelitian ini untuk dicari solusinya. Rumusan masalah yang diangkat pada penelitian ini adalah bagaimana menerapkan sistem pengendali pada sistem *turret gun*, bagaimana mendapatkan nilai K_P dan K_I melalui *neural network* pada sistem pengendali, dan bagaimana meningkatkan respon dari sistem *control* pada *turret gun*

3. Pengambilan data input (error pada sistem) dan output (K_P Azimuth, K_I Azimuth, K_P Elevasi, dan K_I Elevasi) pada sistem

Pengambilan data input dan output diperlukan karena metode *neural network* dibutuhkan pelatihan untuk mendapatkan fungsi dari *gain*. Data input dan output diperoleh dari pengambilan data pada sistem pelontar peluru yang sudah ada [11] dan telah dipasang sistem pengendali. Untuk mendapatkan nilai K_P dan K_I optimum dibutuhkan persamaan yang menyatakan performa dari respon tersebut. Adapun persamaan tersebut adalah *performance index* (F) yang diperoleh dari persamaan [6]:

$$F = (k_1 \times \%Overshoot + k_2 \times RiseTime + k_3 \times steady\ state\ error) \quad (3.1)$$

Dimana:

$$k_1 = 100$$

$$k_2 = 5$$

$$k_3 = 100$$

Performce index tersebut tidak mutlak seperti persamaan 3.1, tetapi dapat diubah seperti yang diinginkan. Pada persamaan diatas adalah menitikberatkan pada *%overshoot* dan *steady-state error*, sehingga performa yang meningkat adalah *%overshoot* dan *steady-state error*. Namun, jika performa respon pada *rise time* yang ingin ditingkatkan, maka nilai k_2 dapat dinaikkan dan atau nilai k_1 dan k_3 diturunkan sehingga nilai k_2 melebihi parameter yang lain. Tetapi jika ingin meningkatkan performa respon pada *settling time*, maka *settling time* ditambahkan pada persamaan diatas dan diberi konstanta pengali.

Pengambilan data dilakukan dengan menyusun kemungkinan error yang dihasilkan oleh sistem. Untuk sudut azimuth terdapat error maksimal yaitu 120, sementara untuk sudut elevasi terdapat error maksimum yaitu 80. Sehingga, untuk sudut azimuth diberikan rentang nilai untuk error dari 0 sampai 120. Kemudian nilai error tersebut dibagi menjadi beberapa nilai, yaitu dibagi menjadi 12 bagian, sehingga penambahan nilai error dari 0 sampai 120 adalah 10. Sedangkan untuk sudut elevasi diberikan rentang

nilai untuk error dari 0 sampai 80. Nilai error elevasi dibagi menjadi 8 bagian, sehingga error untuk sudut elevasi diberi penambahan nilai sebanyak 10.

Masing-masing nilai error tersebut disimulasikan dengan memberikan nilai K_P dan K_I pada masing-masing sudut. Nilai K_P ditentukan pada rentang nilai tiga sampai sepuluh dan nilai K_I ditentukan pada rentang nilai nol sampai tiga. Setiap satu nilai K_P Azimuth, nilai K_P Elevasi ditambah sebanyak 1 nilai. Karakteristik respon yang diambil pada setiap simulasi adalah *%overshoot*, *rise time*, dan *steady-state error*. Kemudian nilai-nilai tersebut dihitung dengan menggunakan persamaan 3.1 sehingga diperoleh data *performance index* (F). Data dari nilai F tersebut kemudian diurutkan pada masing-masing nilai error. Masing-masing nilai error yang mempunyai nilai F paling kecil kemudian diambil untuk dijadikan data input dari pelatihan *neural network*.

Output yang digunakan adalah nilai K_P Azimuth, K_I Azimuth, K_P Elevasi, dan K_I Elevasi. Sama seperti pengambilan data input, data output juga diperoleh dari nilai K_P dan K_I pada masing-masing nilai error yang mempunyai nilai F paling kecil.

4. Membangun Model *Neural network*

Dalam hal membangun model *neural network* terdapat langkah-langkah yang harus dilakukan yaitu sebagai berikut:

a. Menentukan parameter model dari *neural network*

Pada tahap ini, model *neural network* dibangun dengan menentukan parameter-parameter dari model *neural network*. Penentuan berapa *layer* dan *neuron* yang akan dibuat menjadi hal yang utama pada langkah ini. Parameter tersebut mempengaruhi output yang dihasilkan oleh *neural network* apakah sesuai dengan yang diinginkan atau tidak.

b. Membangun model *Neural network*

Pada tahap ini, model *neural network* dibangun dengan memasukkan parameter-parameter dari model *neural network*, yaitu menentukan jumlah *layer* dan *neuron* yang digunakan.

c. Melatih model *neural network*

Pelatihan model *neural network* dimaksudkan untuk mendapatkan nilai K_P dan K_I . Pada langkah ini, data input dan output dimasukkan kedalam *neural network* untuk diproses. Input untuk dimasukkan ke dalam pelatihan *neural network* diperoleh dari data error masing-masing sudut dari sistem yang sudah ada. Sementara output yang dimasukkan ke dalam pelatihan *neural network* adalah nilai K_P Azimuth, K_I Azimuth, K_P Elevasi, dan K_I Elevasi.

d. Mendapatkan parameter *weight* dan *bias*

Output yang dihasilkan dari pelatihan *neural network* adalah *weight* dan *bias*. Kedua parameter tersebut adalah parameter yang mempengaruhi persamaan dari *neural network* untuk menghasilkan nilai K_P dan K_I .

5. Verifikasi model

Untuk menyempurnakan model, dibutuhkan verifikasi yang menyatakan jika model tersebut dapat digunakan. Verifikasi tersebut dilakukan dengan menyamakan hasil akhir dari output sistem dengan set point. Setelah validasi tersebut, maka diperoleh *transient response* dari sistem yang telah diberi pengendali.

6. Kesimpulan dan Saran

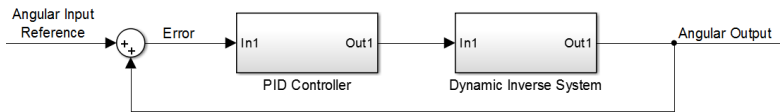
Kesimpulan dari penelitian Tugas Akhir dapat diambil jika pada analisa diperoleh respon yang stabil. Kesimpulan berupa respon kestabilan dan hasil persamaan *proportional gain* (K_P) dan *integral gain* (K_I) melalui *neural network* pada sistem pengendali pelontar peluru plastik. Adapun saran dapat berupa masukan untuk analisa simulasi pelontar peluru plastik pada khususnya.

BAB IV

PENENTUAN PARAMETER GAIN DENGAN *NEURAL NETWORK*

4.1 Pengambilan Data

Pengambilan data dilakukan dengan melakukan simulasi terhadap model dari sistem turret-gun yang telah dilakukan pada penelitian sebelumnya [11]. Blok diagram yang digunakan dalam simulasi adalah sebagai berikut:



Gambar 4.1 Blok diagram sistem *turret-gun* pada penelitian sebelumnya [11]

Pada model sistem yang diperoleh dari penelitian sebelumnya [11], terdapat input referensi berupa sudut yaitu sudut azimuth dan sudut elevasi. Pada sistem tersebut juga terdapat sistem pengendali yaitu *PID controller*. Input yang digunakan untuk *PID controller* adalah error yang terjadi pada sistem. Output dari *controller* adalah torsi. Torsi tersebut digunakan sebagai input untuk sistem *turret-gun*. Output dari sistem adalah sudut azimuth dan sudut elevasi.

Pada blok *PID controller* terdapat persamaan PID. Persamaan tersebut terdapat parameter *proportional gain* (K_P), *Integral gain* (K_I), dan *derivative gain* (K_D). Nilai tersebut adalah faktor pengali untuk nilai error yang terjadi pada sistem. Untuk kebutuhan pengambilan data, K nilai yang diubah adalah nilai K_P dan K_I .

Pengambilan data dilakukan dengan menyusun kemungkinan error yang dihasilkan oleh sistem. Untuk sudut azimuth terdapat error maksimal yaitu 120, sementara untuk sudut elevasi terdapat error maksimum yaitu 80. Sehingga, untuk sudut azimuth diberikan rentang nilai untuk error dari 0 sampai 120. Kemudian nilai error tersebut dibagi menjadi beberapa nilai, yaitu dibagi menjadi 12 bagian, sehingga penambahan nilai error dari 0 sampai 120 adalah 10. Sedangkan untuk sudut elevasi diberikan rentang

nilai untuk error dari 0 sampai 80. Nilai error elevasi dibagi menjadi 8 bagian, sehingga error untuk sudut elevasi diberi penambahan nilai sebanyak 10.

Masing-masing nilai error tersebut disimulasikan dengan memberikan nilai K_P dan K_I pada masing-masing sudut. Nilai K_P ditentukan pada rentang nilai tiga sampai sepuluh dan nilai K_I ditentukan pada rentang nilai nol sampai tiga. Setiap satu nilai K_P Azimuth, nilai K_P Elevasi ditambah sebanyak 1 nilai. Karakteristik respon yang diambil pada setiap simulasi adalah *%overshoot*, *Rise time*, dan *Steady-state Error*. Kemudian nilai-nilai tersebut dihitung dengan menggunakan persamaan 3.1 sehingga diperoleh data *performance index* (F). Data dari nilai F tersebut kemudian urutkan pada masing-masing nilai error. Masing-masing nilai error yang mempunyai nilai F paling kecil kemudian diambil untuk dijadikan data input dari pelatihan *neural network*.

Output yang digunakan adalah nilai K_P Azimuth, K_I Azimuth, K_P Elevasi, dan K_I Elevasi. Sama seperti pengambilan data input, data output juga diperoleh dari nilai K_P dan K_I pada masing-masing nilai error yang mempunyai nilai F paling kecil.

4.2 Hasil Pengambilan Data

Pengambilan data dilakukan dengan menggunakan sistem pada penelitian sebelumnya [11]. Sesuai dengan persamaan 3.1, parameter yang diambil adalah *%overshoot* (OS), *rise time* (RT), dan *steady-state error*. Dengan kombinasi nilai error pada masing-masing sudut, nilai K_P azimuth, K_I azimuth, K_P elevasi, dan K_I elevasi, diperoleh data sebanyak 2038 data. Data tersebut kemudian dipilih sesuai dengan nilai F yang mempunyai nilai terkecil pada setiap nilai error. Sehingga diperoleh data yang ditunjukkan tabel 4.1.

Tabel 4.1 Data untuk pelatihan *neural network* untuk sudut azimuth

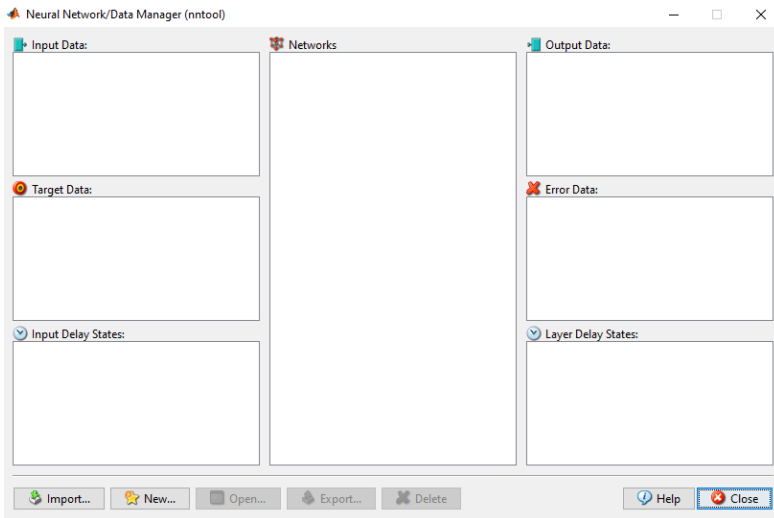
Input (Error Sudut Azimuth)	K _p azimuth	K _i azimuth	OS (%)	RT (detik)	Steady-state Error (derajat)	F
0	3	0	14,596	2,215	0,16384	1487,11
10	3	0	14,528	2,193	0,19752	1483,50
20	3	0	14,707	2,225	0,14070	1495,98
30	3	0	14,613	2,205	0,16590	1488,95
40	3	0	13,908	2,230	0,10285	1412,23
50	3	0	13,641	2,197	0,10882	1385,96
60	3	0	14,412	2,244	0,07993	1460,41
70	3	0	14,235	2,206	0,05596	1440,10
80	3	0	12,571	2,217	0,04288	1272,49
90	3	0	15,298	2,248	0,04025	1545,06
100	3	0	15,559	2,236	0,02683	1569,77
110	3,5	3	0	7,788	13,90202	1429,14
120	0	0	inf	0	0	0

Tabel 4.2 Data untuk pelatihan *neural network* untuk sudut elevasi

Input (Error Sudut Azimuth)	K _p azimuth	K _i azimuth	OS (%)	RT (detik)	Steady-state Error (derajat)	F
0	3,5	0	0,07639	2,40565	0,08993	28,66024
10	5,5	0	0,69529	0,66179	1,76416	249,25399
20	5,5	0	0,12719	2,24329	0,92146	116,08154
30	5	0,5	0,42909	2,09783	0,08367	61,76598
40	10	0	0,09566	0,36071	1,33752	145,12123
50	4	0	0,32753	4,91945	0,12263	69,61268
60	7	1	0,48999	0,48999	1,92975	244,42480
70	4,5	0	0,45566	7,62752	0,76679	160,38270
80	6	0	0,00058	2,37060	1,11535	123,44624

4.3 Pelatihan Neural Network

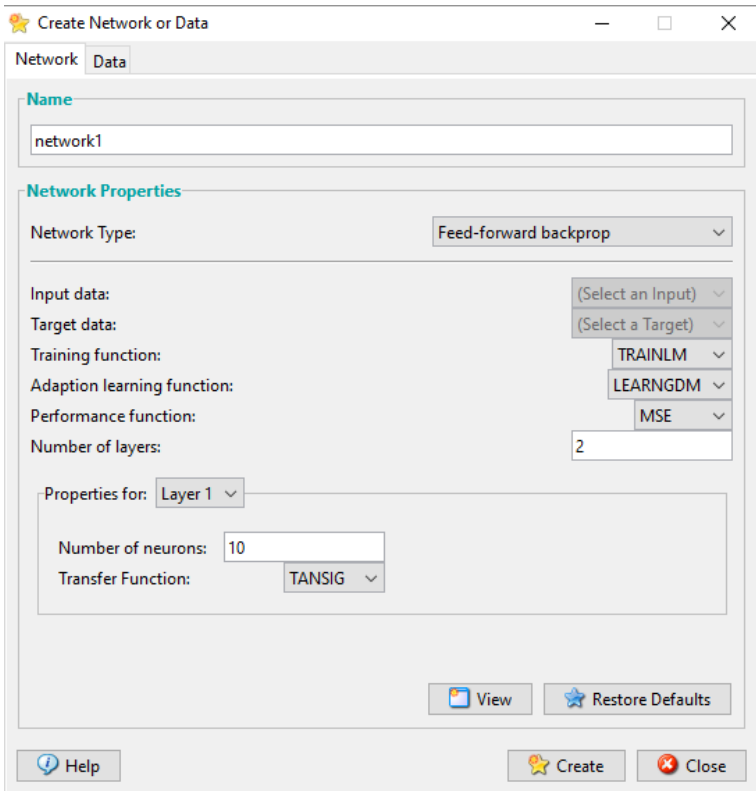
Pelatihan *neural network* dilakukan dengan *software* matlab. Pada *software* matlab terdapat *tool* berupa *data manager* untuk mengambil, membuat, menggunakan, dan memindahkan *networks* dan data. Dengan menggunakan syntax `nntool`, *data manager* dapat dipanggil seperti gambar dibawah ini:



Gambar 4.2 *Toolbox Data Manager*

Langkah selanjutnya yaitu memasukkan data untuk pelatihan *neural network*. Data pada tabel 4.1 dan 4.2 terdapat nilai error dan nilai K_P dan K_I . Nilai error tersebut dimasukkan ke dalam input data, sedangkan nilai K_P dan K_I dimasukkan ke dalam target data.

Untuk membuat *neural network* dilakukan dengan memilih *new* pada *toolbox data manager*. Terdapat beberapa pilihan pada *toolbox create network or data* untuk menentukan model *neural network* yang diinginkan, seperti menentukan tipe dari *neural network*, jumlah *layer* dan *neuron*, dan lain-lain seperti pada gambar 4.2.



Gambar 4.3 Toolbox Create Network or Data

4.2.1 Pelatihan *Neural Network* untuk Sudut Azimuth

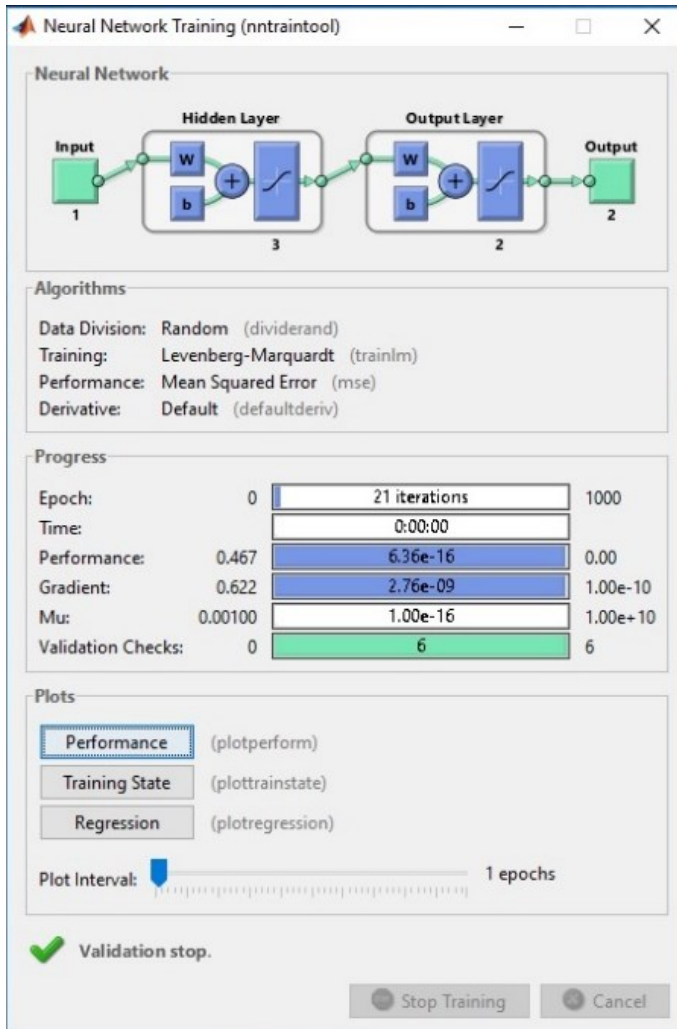
Untuk sudut azimuth digunakan *neural network feed-forward backPropagation* dengan satu input layer, satu hidden layer dan satu output layer. Pada hidden layer terdapat 3 neuron dan pada output layer terdapat 2 neuron. Kombinasi layer tersebut diperoleh dengan trial dan error pada pelatihan *neural network*. Hasil trial dan error adalah sebagai berikut:

Tabel 4.3 Hasil *trial* dan *error Neural Network* sudut azimuth

No	Input Layer	Hidden Layer	Neuron Hidden Layer	Output Layer	Neuron Output Layer	Best MSE
1	1	1	3	1	2	$6,36 \times 10^{-16}$
2	1	1	5	1	2	$3,98 \times 10^{-12}$
3	1	1	10	1	2	$4,65 \times 10^{-12}$

Tabel 4.3 adalah hasil *trial* dan *error* untuk sudut azimuth yang dilakukan sebanyak tiga kali. Percobaan tersebut dilakukan dengan mengubah jumlah *hidden layer* dan jumlah *neuron* pada *hidden layer*. Pada percobaan pertama, diatur *hidden layer* yang berjumlah satu dengan jumlah *neuron* 3. Dari percobaan pertama diperoleh *mean square error* (MSE) sebesar 0,553. Ketika dilakukan simulasi sebanyak 5 kali, MSE terbaik yang diperoleh adalah $6,36 \times 10^{-16}$. Untuk percobaan kedua, jumlah *neuron* pada *hidden layer* ditambah menjadi 5 *neuron* sehingga dihasilkan MSE sebesar $3,98 \times 10^{-12}$. Kemudian untuk percobaan ketiga jumlah *neuron* pada *hidden layer* kembali ditambahkan menjadi 10, sehingga menghasilkan MSE sebesar 0,72.

Dengan kombinasi *layer* dan *neuron* diatas, diperoleh performa terbaik dari *neural network* yang diukur dengan *mean square error* (MSE) yaitu $6,36 \times 10^{-16}$. Adapun parameter-parameter yang membatasi pelatihan *neural network* adalah gradien maksimum yang diatur pada nilai 10^{-10} , jumlah iterasi dibatasi hingga mencapai nilai 1000, dan validasi diatur di angka enam. Hasil dari performa *neural network* setelah pelatihan dapat dilihat pada gambar 4.4.

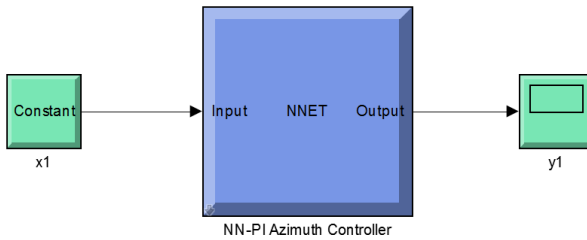


Gambar 4.4 Performa dari *neural network* sudut azimuth setelah pelatihan

Gambar 4.4 adalah hasil pengujian dari *neural network* untuk sudut azimuth. Performa yang diukur adalah *mean square error*.

Setelah menempuh 21 kali iterasi, diperoleh MSE sebesar $6,36 \times 10^{-16}$, gradien sebesar $2,76 \times 10^{-9}$, dan validasi *neural network* sebanyak enam kali.

Hasil dari pelatihan tersebut diperoleh persamaan *neural network*. Didalam persamaan tersebut terdapat nilai *weight* dan *bias*. Pada blok simulink, persamaan *neural network* dimasukkan pada blok seperti dibawah ini:



Gambar 4.5 Blok *Neural network Proportional-Integral* (NN-PI) *controller* untuk sudut azimuth

Blok diatas adalah blok yang dimasukkan ke dalam *controller* pada model sistem turret gun dengan input error dan output nilai K_P dan K_I . Pada blok tersebut terdapat persamaan *neural network* yang terdiri dari nilai input (matriks 13×1), *weight* dan *bias*. Nilai dari *weight* dan *bias* pada *hidden layer* adalah sebagai berikut:

$$w^{(1)} = \begin{bmatrix} 6,84458 \\ -6,99744 \\ 6,99876 \\ 7,02570 \\ -6,22230 \end{bmatrix}$$

$$b^{(1)} = \begin{bmatrix} -7,22267 \\ 3,35407 \\ -0,05692 \\ 3,44517 \\ -7,74226 \end{bmatrix}$$

Dari perhitungan nilai *weight* dan *bias* pada *hidden layer* dihasilkan matriks 5×1 . Perhitungan tersebut diperoleh dari persamaan berikut:

$$z^{(1)} = (\text{input} \times w^{(1)}) + b^{(1)} \quad (4.1)$$

Sedangkan pada *output layer*, *weight* dan *bias neural network* adalah sebagai berikut:

$$w^{(2)} = \begin{bmatrix} 6,98864 \times 10^{-9} & 1,05695 \\ 2,36303 \times 10^{-11} & -0,65778 \\ -2,58803 \times 10^{-12} & 0,16786 \\ 4,96273 \times 10^{-13} & -0,60260 \\ -1,94676 \times 10^{-10} & 1,85312 \end{bmatrix}$$

$$b^{(2)} = [0,89588 \quad -3,51183]$$

Dari perhitungan nilai *weight* dan *bias* pada *output layer* dihasilkan matriks 2×1 . Perhitungan tersebut diperoleh dari persamaan berikut:

$$y = (z^{(1)} \times w^{(2)}) + b^{(2)} \quad (4.2)$$

Sehingga persamaan *neural network controller* untuk sudut azimuth adalah:

$$\begin{bmatrix} k_p \\ k_i \end{bmatrix} = (z^{(1)} \times w^{(2)}) + b^{(2)} \quad (4.3)$$

4.2.2 Pelatihan Neural Network untuk Sudut Elevasi

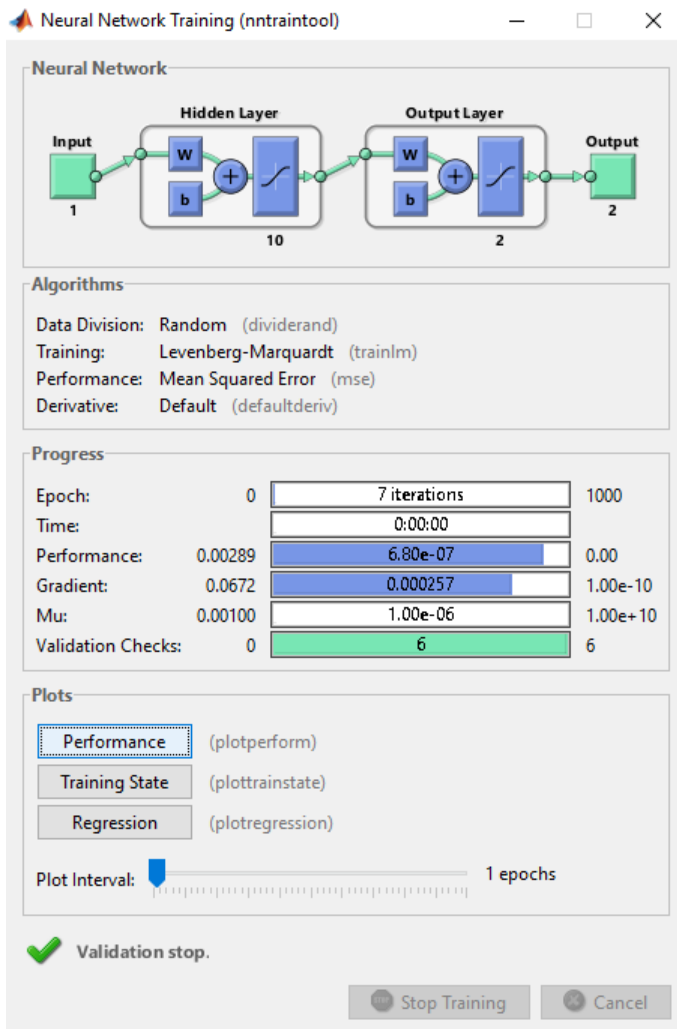
Untuk sudut elevasi digunakan *neural network feed-forward backPropagation* dengan satu *input layer*, satu *hidden layer* dan satu *output layer*. Pada *hidden layer* terdapat 10 *neuron* dan pada *output layer* terdapat 2 *neuron*. Kombinasi *layer* tersebut diperoleh dengan *trial* dan *error* pada pelatihan *neural network*. Hasil *trial* dan *error* adalah sebagai berikut:

Tabel 4.4 Hasil *trial* dan *error Neural Network* untuk sudut elevasi

No	Input Layer	Hidden Layer	Neuron Hidden Layer	Output Layer	Neuron Output Layer	Best MSE
1	1	1	3	1	2	0,304
2	1	1	5	1	2	$6,88 \times 10^{-12}$
3	1	1	10	1	2	$6,8 \times 10^{-12}$

Tabel 4.4 adalah hasil *trial* dan *error* untuk sudut elevasi yang dilakukan sebanyak tiga kali. Percobaan tersebut dilakukan dengan mengubah jumlah *hidden layer* dan jumlah *neuron* pada *hidden layer*. Pada percobaan pertama, diatur *hidden layer* yang berjumlah satu dengan jumlah *neuron* 3. Diperoleh *mean square error* (MSE) terbaik yang diperoleh adalah 0,304. Untuk percobaan kedua, jumlah *neuron* pada *hidden layer* ditambah menjadi 5 *neuron* sehingga dihasilkan MSE sebesar $6,88 \times 10^{-5}$. Kemudian untuk percobaan ketiga jumlah *neuron* pada *hidden layer* kembali ditambahkan menjadi 10, sehingga menghasilkan MSE sebesar $6,8 \times 10^{-7}$.

Dengan kombinasi *layer* dan *neuron* diatas, diperoleh performa terbaik dari *neural network* yang diukur dengan *mean square error* (MSE) yaitu $6,8 \times 10^{-7}$. Adapun parameter-parameter yang membatasi pelatihan *neural network* adalah gradien maksimum yang diatur pada nilai 10^{-10} , jumlah iterasi dibatasi hingga mencapai nilai 1000, dan validasi diatur di angka enam. Hasil dari performa *neural network* setelah pelatihan ditunjukkan pada gambar 4.6.

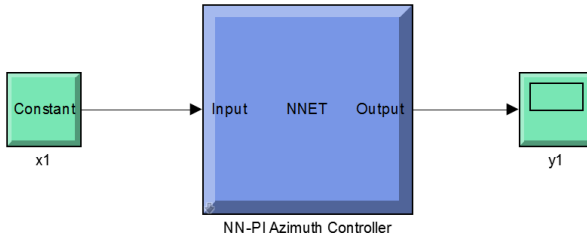


Gambar 4.6 Performa dari *neural network* untuk sudut elevasi setelah pelatihan

Pada gambar 4.6 terdapat hasil pengujian hasil dari *neural network*. Perfoma yang diukur adalah *mean square error*. Setelah

menempuh 29 kali iterasi, diperoleh MSE sebesar $6,8 \times 10^{-7}$, gradien sebesar 0,000257, dan validasi *neural network* sebanyak enam kali.

Hasil dari pelatihan tersebut diperoleh persamaan *neural network*. Didalam persamaan tersebut terdapat nilai *weight* dan *bias*. Pada blok simulink, persamaan *neural network* dimasukkan pada blok seperti dibawah ini:



Gambar 4.7 Blok *Neural network Proportional-Integral* (NN-PI) *controller* untuk sudut elevasi

Blok diatas adalah blok yang dimasukkan ke dalam *controller* pada model sistem turret gun dengan input error dan output nilai K_P dan K_I . Pada blok tersebut terdapat persamaan *neural network* yang terdiri dari nilai input (matriks 13×1), *weight* dan *bias*. Nilai dari *weight* dan *bias* pada *hidden layer* adalah sebagai berikut:

$$w^{(1)} = \begin{bmatrix} -13,71450 \\ -14,19598 \\ 13,65194 \\ 13,94620 \\ 13,98112 \\ 14,03254 \\ 14,02302 \\ -13,99036 \\ 13,66881 \\ -13,99974 \end{bmatrix}$$

$$b^{(1)} = \begin{bmatrix} 14,28410 \\ 10,62672 \\ -8,49205 \\ -5,06125 \\ -1,73043 \\ 1,76700 \\ 4,57454 \\ -7,79051 \\ 1,33047 \\ -14,00035 \end{bmatrix}$$

Dari perhitungan nilai *weight* dan *bias* pada *hidden layer* dihasilkan matriks 5×1 . Perhitungan tersebut diperoleh dari persamaan berikut:

$$z^{(1)} = (\text{input} \times w^{(1)}) + b^{(1)} \quad (4.4)$$

Sedangkan pada *output layer*, *weight* dan *bias neural network* adalah sebagai berikut:

$$w^{(2)} = \begin{bmatrix} -1,03570 & -1,13429 \\ -0,15286 & 0,05016 \\ -0,59520 & -2,88941 \\ 0,79812 & 2,78544 \\ -2,56353 & 1,29839 \\ 2,21899 & -0,00428 \\ -0,12393 & -0,72511 \\ 0,33191 & 0,54962 \\ -0,42312 & -1,39942 \\ 0,48068 & 0,69317 \end{bmatrix}$$

$$b^{(2)} = [1,64723 \quad 0,24712]$$

Dari perhitungan nilai *weight* dan *bias* pada *output layer* dihasilkan matriks 2×1 . Perhitungan tersebut diperoleh dari persamaan berikut:

$$y = (z^{(1)} \times w^{(2)}) + b^{(2)} \quad (4.5)$$

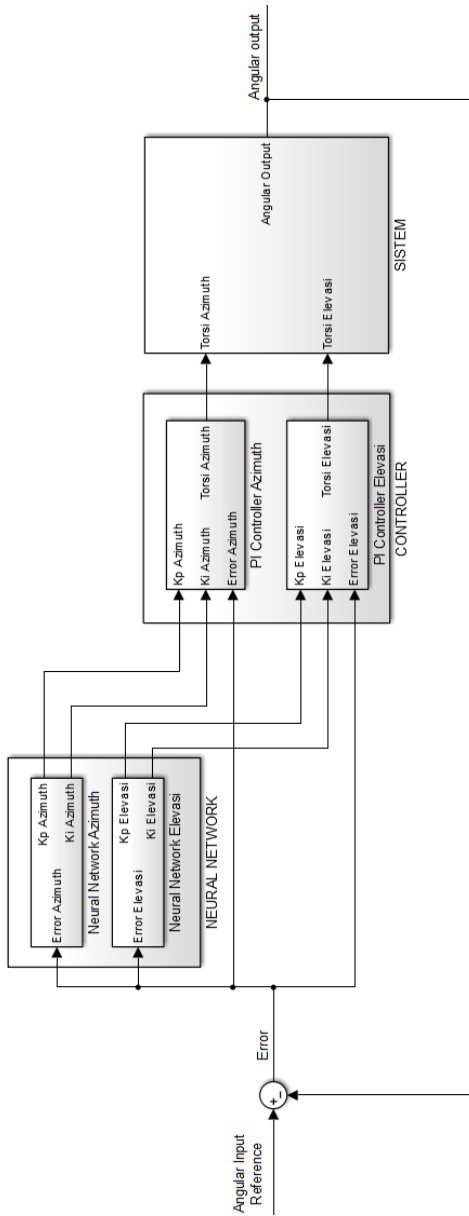
Sehingga persamaan *neural network controller* untuk sudut elevasi adalah:

$$\begin{bmatrix} k_p \\ k_i \end{bmatrix} = (z^{(1)} \times w^{(2)}) + b^{(2)} \quad (4.6)$$

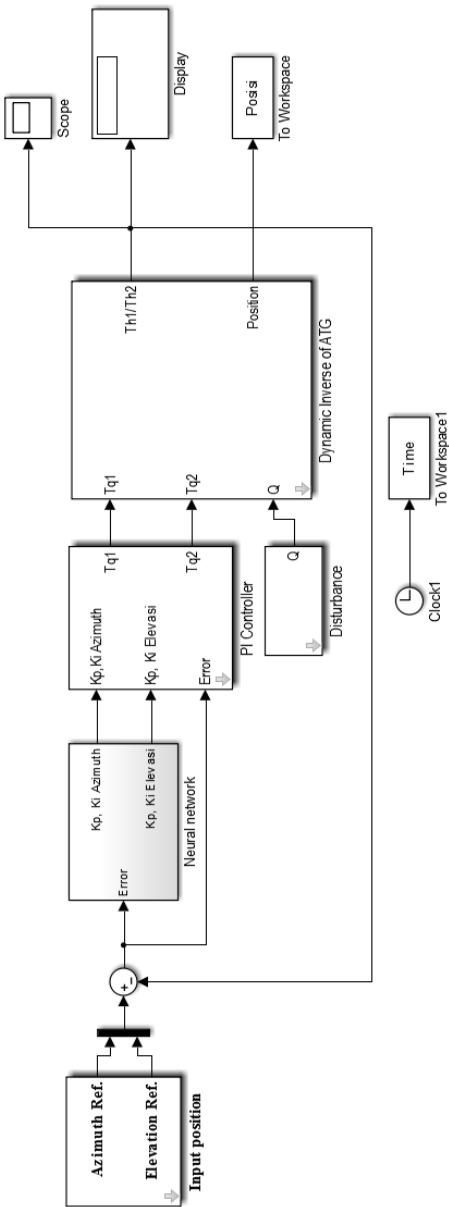
4.4 Blok Diagram dan Model Simulink dari Sistem *Turret-Gun* serta Sistem Pengendalinya

Setelah didapatkan persamaan neural network untuk sudut azimuth dan elevasi, maka keseluruhan komponen untuk membuat model sistem *turret-gun* dengan *Neural Network Proportional Integral* (NN-PI) *controller* telah diperoleh. Model sistem tersebut selanjutnya disimulasikan sesuai dengan kriteria yang ada dengan tujuan untuk mengetahui apakah sistem tersebut telah sesuai dengan yang diinginkan.

Gambar 4.8 menunjukkan proses pada sistem dari input hingga output. Blok diagram tersebut terdiri dari input, *controller*, *neural network*, sistem, dan output. Input sistem berupa sudut yang diinginkan, yaitu sudut azimuth dan sudut elevasi. Pada blok *controller* berisi persamaan *proportional* dan *integral* control. Blok *controller* mempunyai input error dan nilai K_p dan K_i untuk masing-masing sudut. Nilai K_p dan K_i didapatkan dari proses perhitungan pada blok *neural network*. Blok *neural network* berisi persamaan yang menghasilkan nilai K_p dan K_i . Dengan memasukkan nilai error yang terjadi pada sistem, maka nilai K_p dan K_i dapat ditemukan. Nilai K_p dan K_i tersebut setelah keluar dari blok *neural network* akan digunakan sebagai factor pengali untuk menghasilkan torsi dari *controller*. Torsi tersebut adalah output dari *controller* yang menjadi input untuk sistem *turret-gun*. Sistem yang digunakan pada tugas akhir ini diperoleh dari penelitian sebelumnya [11]. Kemudian sebelum menuju output, keseluruhan sistem diberi *feedback* untuk faktor koreksi. Setelah itu, output yang dihasilkan mampu mendekati nilai sebenarnya. Setelah diperoleh *neural network* dan *proportional-integral control* yang sesuai, maka metode *neural network* berhasil dilakukan.



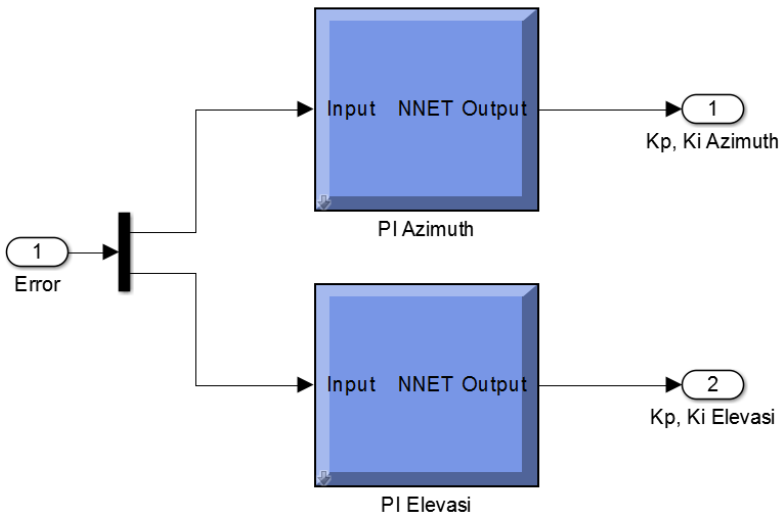
Gambar 4.8 Blok diagram sistem turret-gim dengan Neural Network Proportional-Integral controller



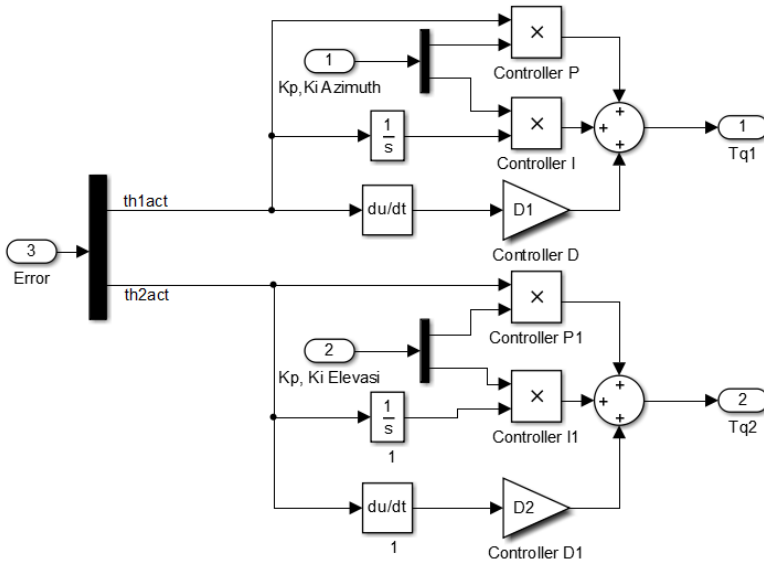
Gambar 4.9 Blok Simulink untuk sistem turret gun dengan NN-PID controller

Blok diagram pada gambar 4.8 diterjemahkan ke dalam software Simulink untuk memudahkan dalam simulasi. Model Simulink sistem turret-gun adalah sebagai berikut:

Gambar 4.9 adalah model Simulink dari sistem turret gun dan sistem pengendalnya. Sistem pengendali yang digunakan adalah *neural network proportional integral*. Sistem terdiri dari input referensi yaitu sudut azimuth dan sudut elevasi. Dari input referensi kemudian dicari error dari sistem. Kemudian nilai error tersebut masuk ke dalam neural network dan sistem pengendali *proportional integral*. Pada blok neural network, nilai error digunakan untuk mencari nilai K_P dan K_I . Penjabaran dari blok *neural network* dan sistem pengendali adalah sebagai berikut:

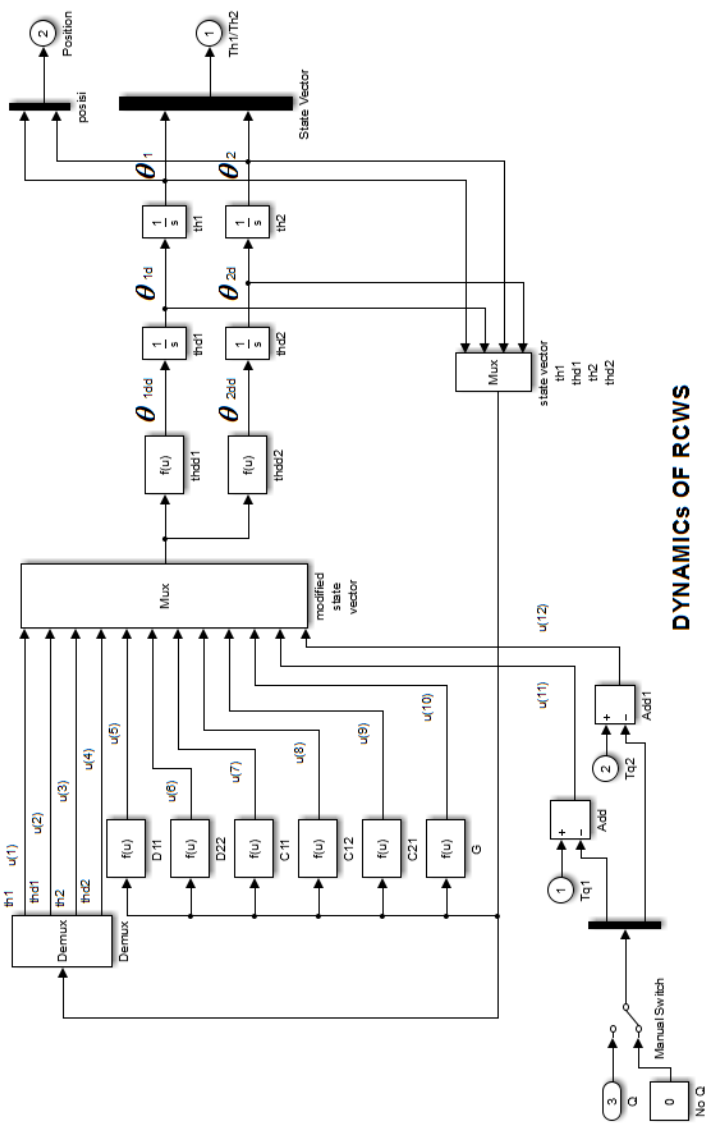


Gambar 4.10 Blok neural network pada model Simulink



Gambar 4.11 Blok Simulink untuk sistem pengendali turret gun

Input 1 pada gambar 4.10 dan input 3 pada gambar 4.11 adalah nilai error yang terjadi pada sistem. Gambar 4.10 menunjukkan blok neural network untuk masing-masing sudut. Blok neural network tersebut menghasilkan output berupa nilai K_P dan K_I untuk sudut azimuth dan sudut elevasi. Output dari neural network tersebut digunakan sebagai input pada sistem pengendali sebagai factor pengali yaitu proportional gain dan integral gain. Sehingga, output yang diperoleh dari sistem pengendali adalah torsi. Torsi tersebut digunakan sebagai input untuk sistem dari turret gun. Sistem turret gun yang digunakan adalah sistem yang telah dibuat pada penelitian sebelumnya [11]. Persamaan model sistem turret gun yang pada persamaan (2.10) dan (2.11) diterjemahkan kedalam model Simulink sebagai berikut:



DYNAMICS OF RCWS

Gambar 4.12 Model Simulink untuk sistem turret gun

[Halaman ini sengaja dikosongkan]

BAB V

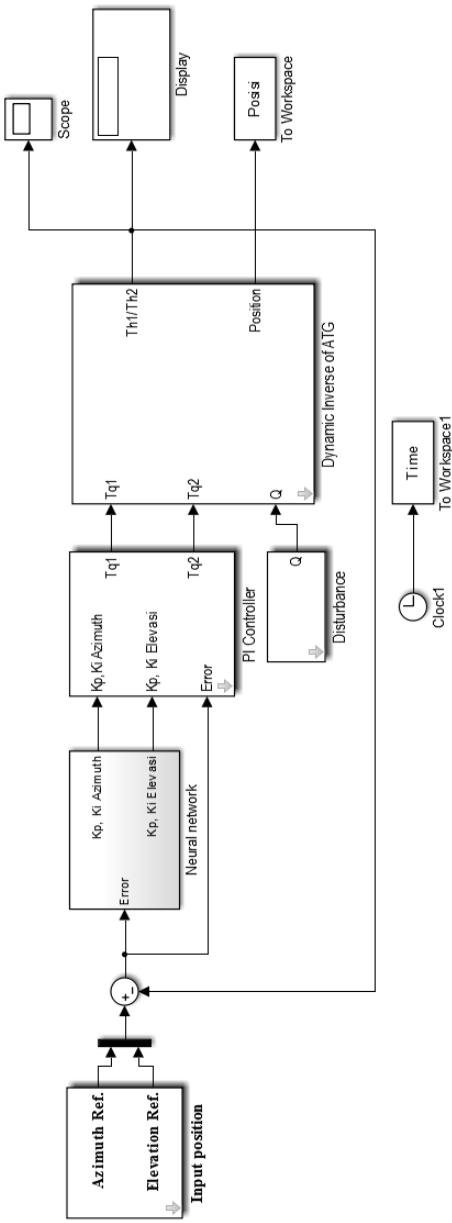
SIMULASI SISTEM DENGAN PROPORTIONAL- INTEGRAL CONTROLLER

5.1 Simulasi

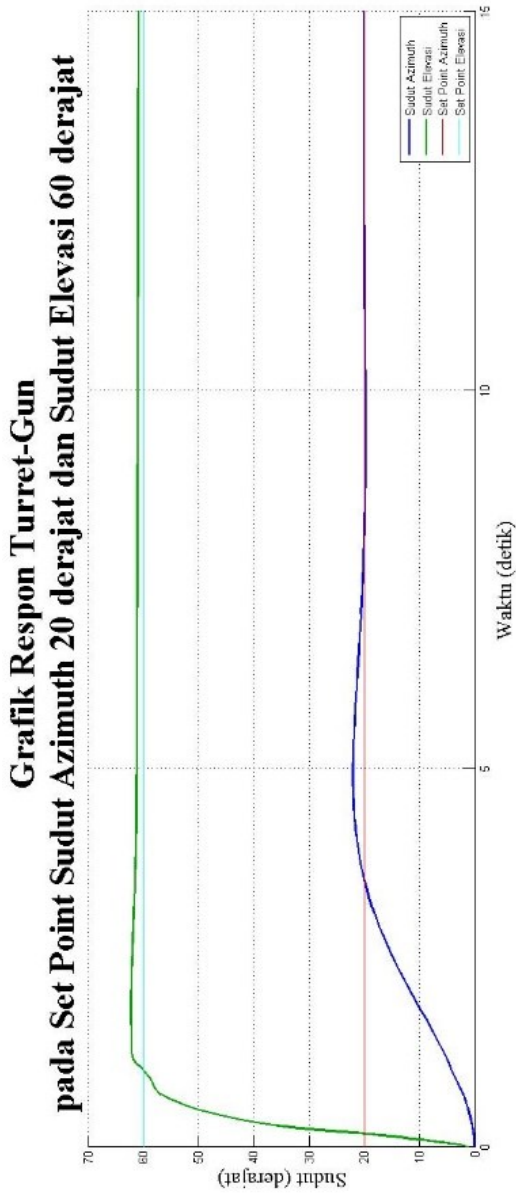
Simulasi dari sistem turret gun dengan *Nerual Network Proportional-Integral* (NN-PI) *controller* dibuat menggunakan software Simulink. Sistem dibuat *closed loop* dengan input *set point* sudut azimuth dan sudut elevasi dan output berupa sudut azimuth dan sudut elevasi aktual. Sistem pengendali menggunakan NN-PI *controller* yang telah dijelaskan sebelumnya. Blok Simulink untuk sistem turret-gun dengan sistem pengendali NN-PI ditunjukkan pada gambar 5.1.

Simulasi dilakukan dengan waktu 10 detik. *Set point* masing-masing sudut ditentukan dari -60 sampai 60 derajat untuk sudut azimuth dan -20 sampai 60 derajat untuk sudut elevasi. Hasil simulasi tersebut ditunjukkan pada gambar 5.2

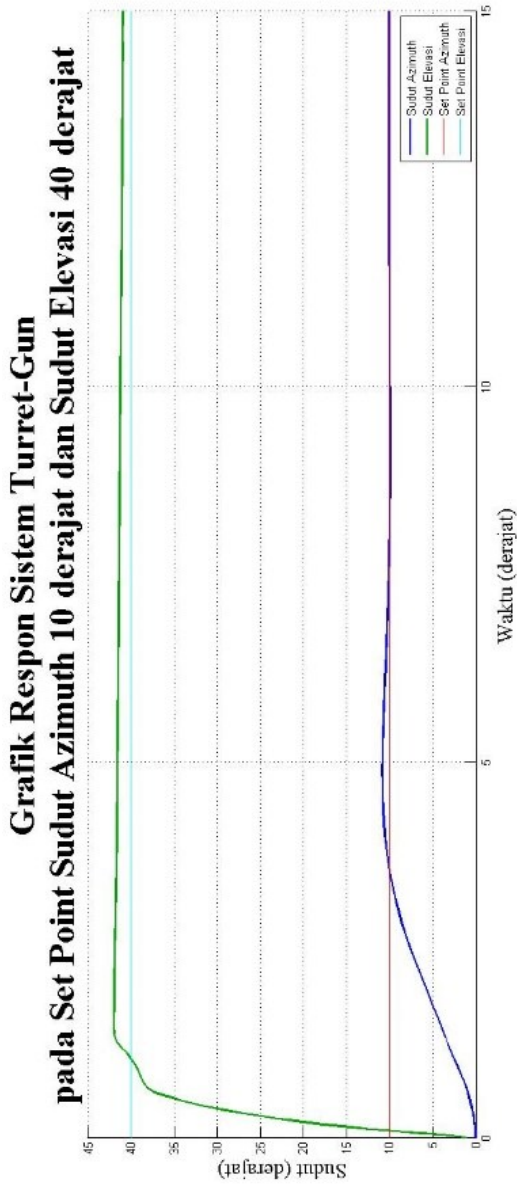
Pada gambar 5.2 diberikan *set point* untuk sudut azimuth sebesar 20 derajat dan sudut elevasi 60 derajat. Dapat dilihat bahwa respon dari sistem telah mampu mendekati nilai aktual yang diinginkan. Respon tersebut memiliki *settling time* sebesar 7,23885 detik untuk sudut azimuth dan untuk sudut elevasi sebesar 2,48108 detik. Untuk *overshoot* dari sistem adalah 9,99398 % untuk sudut azimuth dan 2,21586 % untuk sudut elevasi. Namun respon dari sistem selama 15 detik masih terdapat error sebesar 0,01480 untuk sudut azimuth dan 0,81172 untuk sudut elevasi.



Gambar 5.1 Sistem Turret-Gun setelah diberikan sistem pengendali



Gambar 5.2 Respon sistem dengan sudut azimuth 20 derajat dan sudut elevasi 60 derajat



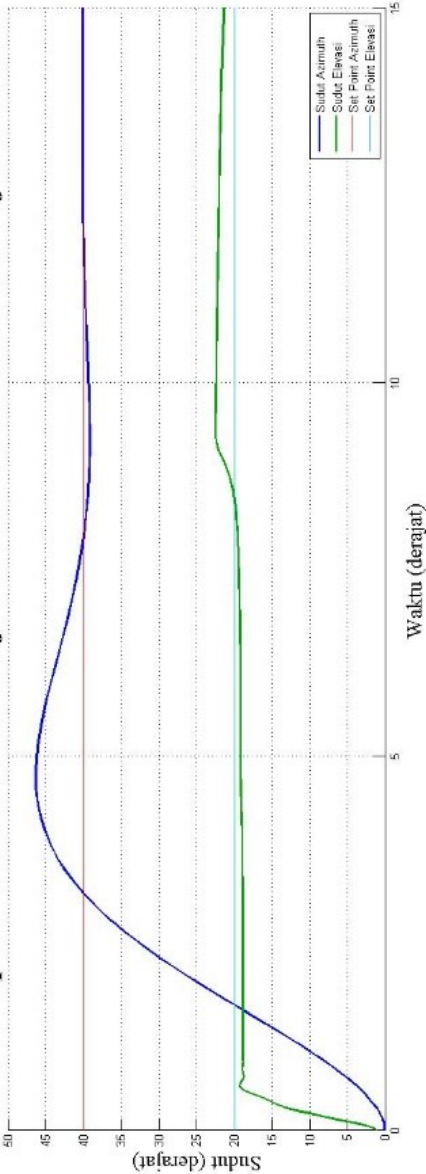
Gambar 5.3 Respon sistem dengan sudut azimuth 10 derajat dan sudut elevasi 40 derajat

Pada gambar 5.3 diberikan set point untuk sudut azimuth sebesar 10 derajat dan sudut elevasi 40 derajat. Dapat dilihat bahwa respon dari sistem telah mampu mendekati nilai actual yang diinginkan. Respon tersebut memiliki *settling time* sebesar 6,98184 detik untuk sudut azimuth dan 3,68276 detik untuk sudut elevasi. Untuk *overshoot* dari sistem adalah 8,15303 % untuk sudut azimuth dan 2,66060 % untuk sudut elevasi. Namun respon dari sistem selama 15 detik masih terdapat error sebesar $3,19984 \times 10^{-3}$ untuk sudut azimuth dan 0,88107 untuk sudut elevasi.

Pada gambar 5.4 diberikan set point untuk sudut azimuth sebesar 40 derajat dan sudut elevasi 20 derajat. Dapat dilihat bahwa respon dari sistem telah mampu mendekati nilai actual yang diinginkan. Respon tersebut memiliki *settling time* sebesar 9,90074 detik untuk sudut azimuth dan 13,74636 detik untuk sudut elevasi. Untuk *overshoot* dari sistem adalah 15,58889 % untuk sudut azimuth dan 5,26888 % untuk sudut elevasi. Namun respon dari sistem masih terdapat *steady state error* sebesar 0,03917 untuk sudut azimuth dan 1,31759 untuk sudut elevasi.

Pada pengujian diberikan beberapa variasi *set point* untuk sudut azimuth dan sudut elevasi. Dengan membandingkan dengan set point sudut azimuth pada nilai -60 sampai 60 derajat, output yang diberikan sistem terdapat error rata-rata sebesar 0,0393. Sedangkan untuk set point sudut elevasi pada nilai -20 sampai 60 derajat dengan output sudut elevasi terdapat error rata-rata sebesar 1,1622.

**Grafik Respon Sistem Turret-Gun
pada Set Point 40 derajat dan Sudut Elevasi 20 derajat**



Gambar 5.4 Respon sistem dengan sudut azimuth 40 derajat dan sudut elevasi 20 derajat

Tabel 5.1 Error posisi rata-rata sistem turret gun pada simulasi selama 15 detik

Set Point Sudut Azimuth (derajat)	Set Point Sudut Elevasi (derajat)	Error Sudut Azimuth (derajat)	Error Sudut Elevasi (derajat)
5	10	0,00247	0,40300
15	35	0,00983	0,61743
20	45	0,02006	1,36889
30	60	0,02532	1,08862
35	10	0,02524	0,70324
40	15	0,02125	1,09741
45	40	0,02753	1,33297
50	10	0,04773	0,74650
55	45	0,02207	1,82759
60	60	0,09845	1,59086
<i>Root Mean Square Error</i>		0,0393063	1,1621708

5.2 Perbandingan PID Controller dengan Neural Network PI Controller

Pengujian juga dilakukan untuk membandingkan performa dari sistem *turret-gun* dengan *PID controller* dan sistem *turret-gun* dengan *NN-PI controller*. Pengujian dilakukan dengan memasukkan set poin sudut azimuth dan elevasi secara acak. Parameter yang diambil untuk membandingkan performa dari sistem adalah *%overshoot(OS)*, *rise time(RT)*, *settling time(ST)*, *steady-state error*, dan *index performance (F)*. Hasil pengujian ditunjukkan pada tabel 5.2

Tabel 5.2 Performa sistem *turret-gun* dengan PID *controller* untuk sudut azimuth

Set Point Sudut Azimuth (derajat)	Set Point Sudut Elevasi (derajat)	OS (%)	RT (detik)	ST (detik)	Steady-state Error (derajat)	F
5	10	37,337	1,018	8,202	0,004351	3739,2
15	35	31,721	1,049	8,323	0,013272	3178,63
20	45	35,372	1,104	8,679	0,028794	3545,55
30	60	33,901	1,104	8,849	0,032954	3398,89
35	10	36,753	1,055	8,734	0,034073	3683,99
40	15	36,961	1,068	8,877	0,044723	3705,91
45	40	36,475	1,076	8,845	0,054643	3658,3
50	10	36,774	1,056	9,011	0,060779	3688,7
55	45	36,443	1,073	8,667	0,088981	3658,55
60	60	36,131	1,066	8,878	0,048178	3623,19
Rata-rata		35,7865	1,0670	8,7066	0,041075	3588,09

Tabel 5.2 adalah hasil simulasi untuk mengetahui performa dari sistem untuk sudut azimuth yang menggunakan sistem pengendali PID. Dengan pengujian pemberian set poin secara acak, didapatkan *%overshoot* rata-rata adalah 35,786 %. Sedangkan untuk *rise time* rata-rata dari sistem adalah 1,067 detik. Kemudian untuk *settling time* rata-rata adalah 8,707 detik. Untuk *steady-state error* rata-rata adalah 0,041075 derajat dan *index performance* sebesar 3588,09.

Tabel 5.3 Performa sistem *turret-gun* dengan PID *controller* untuk sudut elevasi

Set Point Sudut Azimuth (derajat)	Set Point Sudut Elevasi (derajat)	OS (%)	RT (detik)	ST (detik)	Steady-state Error (derajat)	F
5	10	0,630	2,677	2,886	0,747411	151,078
15	35	0,168	2,877	5,396	0,846432	115,776
20	45	1,180	2,213	2,574	1,909916	320,056
30	60	0,094	2,513	8,283	0,94521	116,523
35	10	2,116	5,938	6,774	0,747416	315,994
40	15	2,872	2,594	4,778	2,818987	582,014
45	40	1,598	2,604	2,843	2,745767	447,364
50	10	0,782	8,203	8,425	0,747456	194,011
55	45	1,195	2,764	5,099	1,909769	324,328
60	60	0,115	2,952	9,233	0,94522	120,806
Rata-rata		1,07491	3,5336	5,6292	1,436358	268,795

Tabel 5.3 adalah hasil simulasi untuk mengetahui performa dari sistem untuk sudut azimuth yang menggunakan sistem pengendali PID. Dengan pengujian pemberian set poin secara acak, didapatkan *%overshoot* rata-rata adalah 1,075 %. Sedangkan untuk *rise time* rata-rata dari sistem adalah 3,534 detik. Kemudian untuk *settling time* rata-rata adalah 5,629 detik. Untuk *steady-state error* rata-rata adalah 1,43636 derajat dan *index performance* sebesar 268,795.

Tabel 5.4 Performa sistem *turret-gun* dengan NN-PI *controller* untuk sudut azimuth

Set Point Sudut Azimuth (derajat)	Set Point Sudut Elevasi (derajat)	OS (%)	RT (detik)	ST (detik)	Steady-state Error (derajat)	F
5	10	10,086	2,211	6,904	0,00248	1019,92
15	35	10,061	2,352	7,129	0,00981	1018,82
20	45	15,629	2,174	7,309	0,02011	1575,81
30	60	10,408	2,178	7,001	0,02533	1054,28
35	10	12,686	2,172	6,855	0,02536	1281,96
40	15	15,520	2,161	7,013	0,0213	1564,88
45	40	15,207	2,185	7,083	0,02765	1534,35
50	10	14,652	2,176	7,017	0,04729	1480,8
55	45	13,643	2,172	6,907	0,02178	1377,35
60	60	12,754	2,178	7,147	0,09847	1296,1
Rata-rata		13,0645	2,1959	7,0364	0,02996	1320,43

Tabel 5.4 adalah hasil simulasi untuk mengetahui performa dari sistem untuk sudut azimuth yang menggunakan sistem pengendali NN-PI. Dengan pengujian pemberian set poin secara acak, didapatkan *%overshoot* rata-rata adalah 13,064 %. Sedangkan untuk *rise time* rata-rata dari sistem adalah 2,196 detik. Kemudian untuk *settling time* rata-rata adalah 7,036 detik. Untuk *steady-state error* rata-rata adalah 0,041075 derajat dan *index performance* sebesar 1320,43.

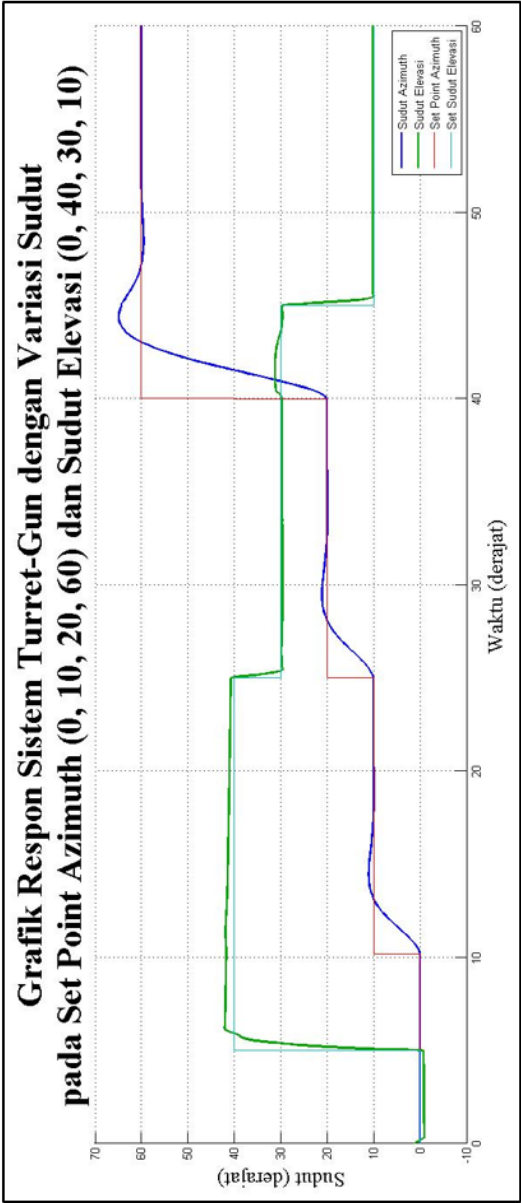
Tabel 5.5 Performa sistem *turret-gun* dengan NN-PI *controller* untuk sudut elevasi

Set Point Sudut Azimuth (derajat)	Set Point Sudut Elevasi (derajat)	OS (%)	RT (detik)	ST (detik)	Steady-state Error (derajat)	F
5	10	3,514	0,421	4,329	0,40296	393,78
15	35	0,569	2,686	2,744	0,61724	132,083
20	45	8,995	0,462	12,144	1,3187	1033,7
30	60	1,437	0,569	3,435	1,08824	255,329
35	10	5,073	3,655	8,126	0,70228	595,782
40	15	4,445	4,086	10,042	1,09778	574,67
45	40	1,914	0,592	4,252	1,32824	327,134
50	10	4,812	3,996	8,428	0,74599	575,832
55	45	3,618	3,672	9,701	1,81146	561,263
60	60	2,018	3,658	4,186	1,59278	379,407
Rata-rata		3,63942	2,3797	6,73859	1,07057	482,898

Tabel 5.5 adalah hasil simulasi untuk mengetahui performa dari sistem untuk sudut azimuth yang menggunakan sistem pengendali NN-PI. Dengan pengujian pemberian set poin secara acak, didapatkan *%overshoot* rata-rata adalah 3,639 %. Sedangkan untuk *rise time* rata-rata dari sistem adalah 2,3797 detik. Kemudian untuk *settling time* rata-rata adalah 6,739 detik. Untuk *steady-state error* rata-rata adalah 1,07057 derajat dan *index performance* sebesar 482,898.

5.3 Simulasi Sistem dengan Variasi Sudut

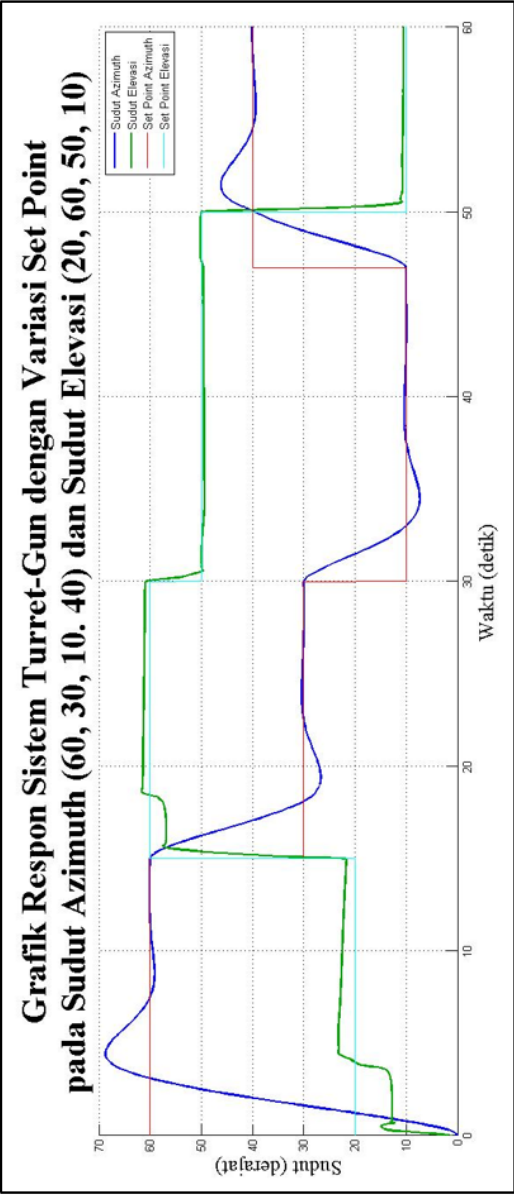
Simulasi sistem dengan variasi sudut diperlukan untuk mengetahui apakah hasil dari neural networks dapat mengikuti perubahan sudut pada setiap waktu. Pengujian sistem turret gun dilakukan dengan variasi sudut secara acak.



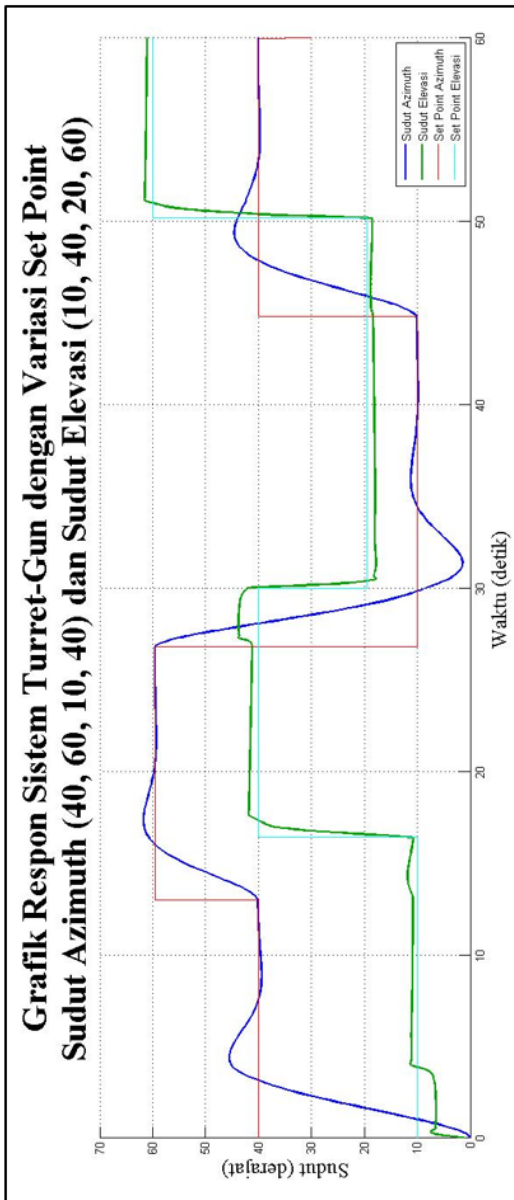
Gambar 5.5 Grafik respon sistem *turret-gun* dengan variasi set point sudut azimuth (0, 10, 20, 60) dan sudut elevasi (0, 40, 30, 10)

Gambar 5.5 adalah grafik respons dari sistem *turret-gun* dengan berbagai variasi sudut. Sudut azimuth diawali dengan sudut 0 derajat. Kemudian pada detik ke 10, set poin sudut azimuth dinaikkan menjadi 10 derajat. Setelah itu, set poin sudut azimuth dinaikkan kembali menjadi 20 derajat pada detik ke 25. Pada set poin terakhir, sudut azimuth dinaikkan kembali menjadi 60 derajat. Sedangkan untuk sudut elevasi diawali dengan set poin 0 derajat. Pada detik ke 5, set poin sudut dinaikkan menjadi 40 derajat. Selang waktu 20 detik, set poin kembali diturunkan menjadi 30 derajat. Terakhir, set poin diturunkan menjadi 10 derajat pada detik ke 45 hingga simulasi selesai pada detik ke 60.

Gambar 5.6 adalah grafik respons dari sistem *turret-gun* dengan variasi set poin. Set poin yang digunakan untuk sudut azimuth secara berurutan adalah 60, 30, 10, dan 40 derajat. Sementara untuk sudut elevasi adalah 20, 60, 50, dan 10 derajat. Simulasi pada sudut azimuth diawali dengan set poin 60 derajat. Setelah 15 detik, set poin diturunkan menjadi 30 derajat. Kemudian set poin diturunkan menjadi 10 derajat pada detik ke 30 dan dinaikkan kembali menjadi 40 derajat pada detik ke 47. Sedangkan untuk sudut elevasi, set poin awal adalah 20 derajat. Pada detik ke 15, set poin dinaikkan menjadi 60 derajat. Kemudian diturunkan menjadi 50 derajat setelah 15 detik. Terakhir, dari detik ke 50 hingga 60, set poin ditetapkan pada angka 10 derajat.



Gambar 5.6 Grafik respon sistem *turret-gun* dengan variasi set point sudut azimuth (60, 30, 10, 40) dan sudut elevasi (20, 60, 50, 10)



Gambar 5.7 Grafik respon sistem *turret-gun* dengan variasi set point sudut azimuth (40, 60, 10, 40) dan sudut elevasi (10, 40, 20, 60)

Gambar 5.7 menunjukkan grafik respons dari sistem *turret-gun* dengan variasi set poin. Untuk sudut azimuth, set poin yang diberikan adalah 40, 60, 10, dan 40 derajat dan untuk sudut elevasi, set poin yang diberikan adalah 10, 40, 20, dan 60. Set poin awal untuk azimuth adalah 40 derajat hingga detik ke 13. Selanjutnya set poin dinaikkan menjadi 60 derajat hingga detik ke 27. Pada detik ke 27, set poin diturunkan menjadi 10 derajat. Setelah itu, set poin kembali dinaikkan menjadi 40 derajat hingga simulasi selesai. Sedangkan untuk sudut elevasi, set awal yang diberikan adalah 10 derajat. Setelah 16 detik, set poin dinaikkan menjadi 40 derajat. Setelah 14 detik, set poin pun kembali diturunkan menjadi 20 derajat. Pada bagian akhir simulasi yaitu pada detik ke 50 sampai 60, set poin dinaikkan menjadi 60 derajat.

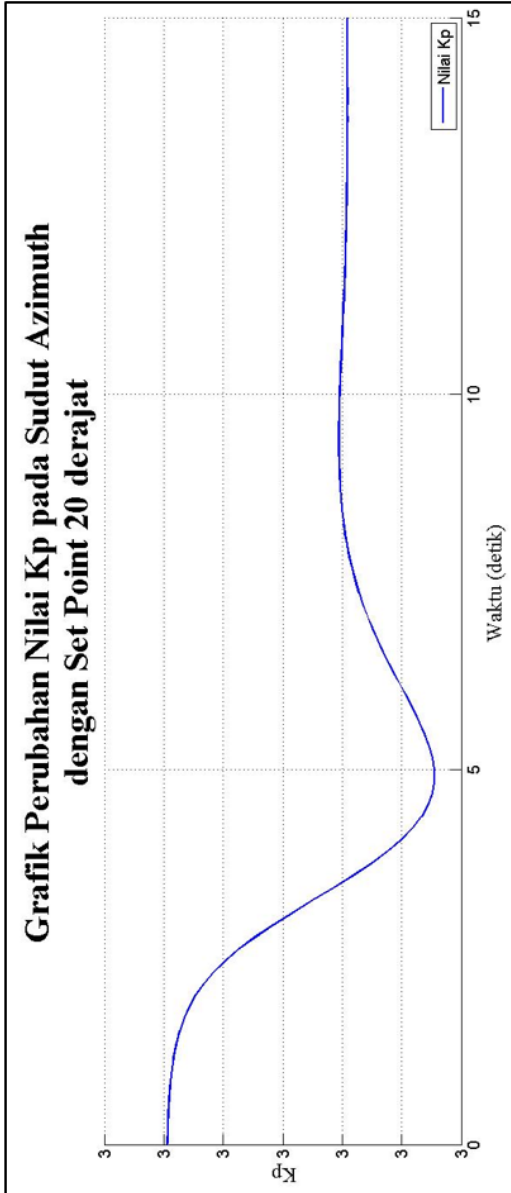
5.4 Parameter Gain Proportional-Integral (PI) pada *Controller*

Parameter *proportional* gain (K_P) dan *integral* gain (K_I) pada *controller* yang menggunakan *neural network* diterapkan sehingga nilai kedua parameter tersebut dapat berubah setiap waktu. *Controller* dapat menyesuaikan nilai K_P dan K_I pada masing-masing error tertentu.

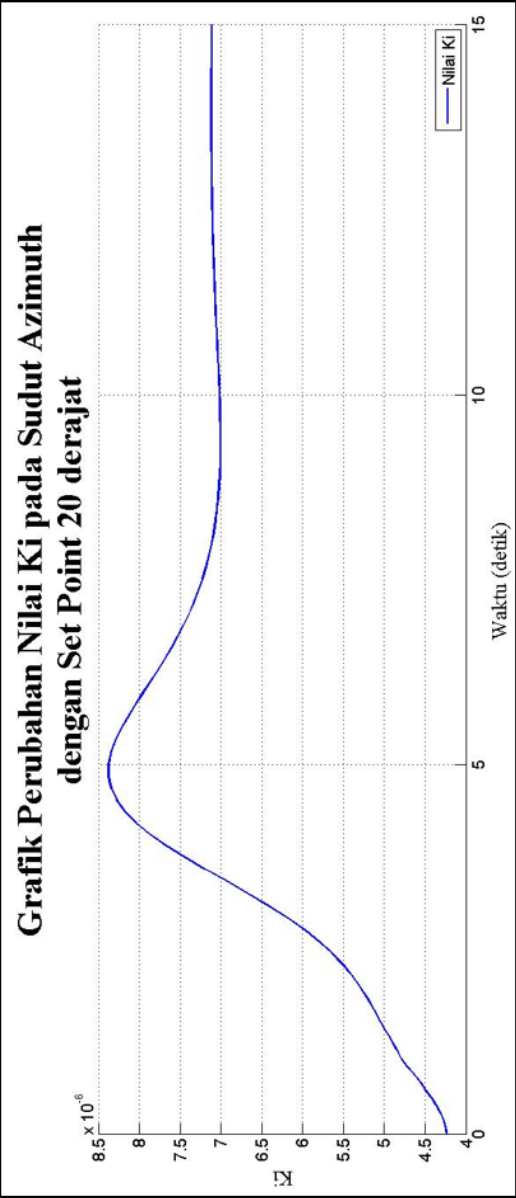
Gambar 5.8 terdapat grafik nilai K_P sudut azimuth pada *set point* sudut azimuth sebesar 20 derajat. Grafik menunjukkan bahwa nilai K_P untuk sudut azimuth berkisar di angka 3. Perubahan pada nilai K_P hanya pada nilai desimalnya saja.

Gambar 5.9 menunjukkan grafik perubahan nilai K_I sudut azimuth dengan setpoint sudut azimuth sebesar 20 derajat. Dapat dilihat bahwa nilai maksimum dari K_I adalah 8.5×10^{-6} . Nilai K_I berada di kisaran nilai 0. Perubahan nilai K_I pada sudut azimuth hanya pada angka desimalnya saja.

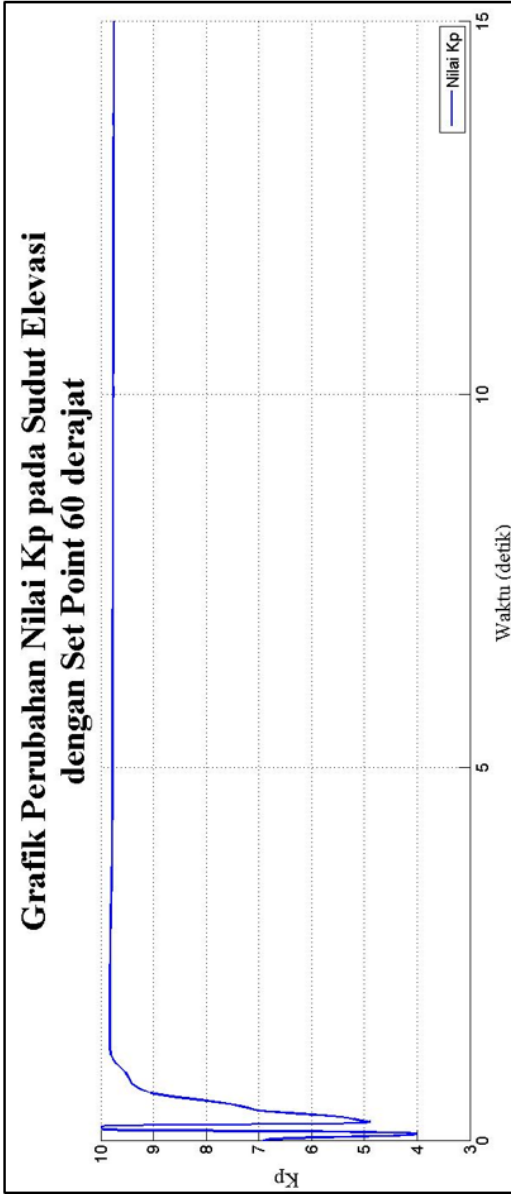
Gambar 5.10 menunjukkan grafik nilai K_P elevasi pada sudut elevasi 60 derajat. Nilai dari K_P elevasi naik turun di awal, namun menuju ke stabil di angka 9. Pada saat K_P memiliki nilai 9 pun masih terdapat perubahan di angka desimalnya.



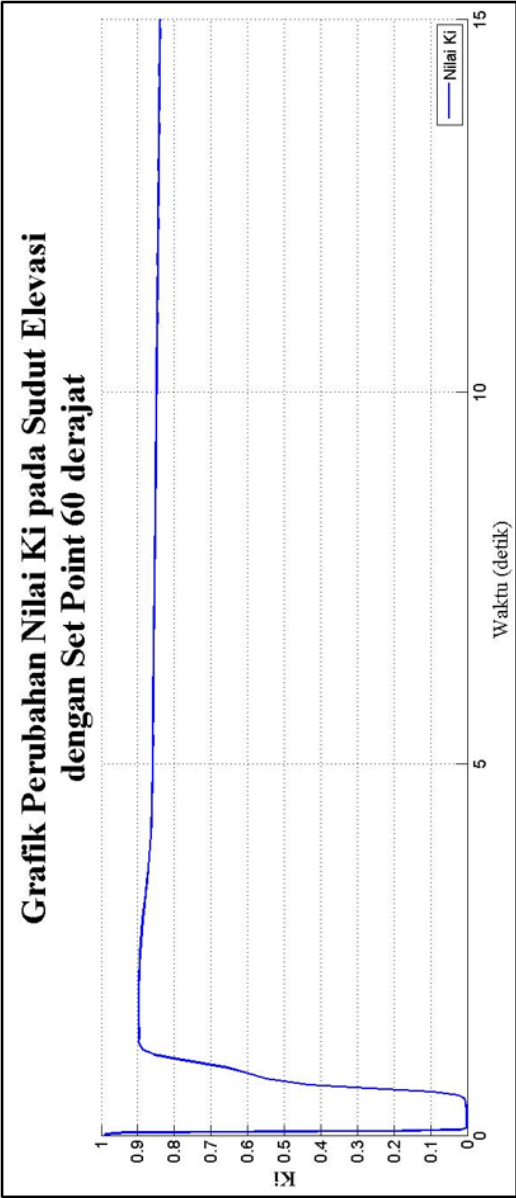
Gambar 5.8 Grafik nilai K_p Azimuth pada sudut azimuth 20 derajat



Gambar 5.9 Grafik nilai K_i azimuth pada sudut azimuth 20 derajat



Gambar 5.10 Grafik nilai K_p elevasi pada sudut elevasi 60 derajat



Gambar 5.11 Grafik nilai K_I elevasi pada sudut elevasi 60 derajat

Gambar 5.11 meunjukkan grafik perubahan nilai K_1 sudut elevasi dengan set point sudut elevasi 60 derajat. Nilai K_1 sudut elevasi berkisar antara 0 sampai 1. Perubahan nilai K_1 sudut elevasi hanya pada angka desimalnya saja.

[Halaman ini sengaja dikosongkan]

BAB VI KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan hasil analisa yang telah dijelaskan pada bab-bab sebelumnya, maka dapat disimpulkan bahwa:

1. Sistem turret gun dapat diberikan *controller* dengan menggunakan metode *neural network*. Berdasarkan hasil simulasi diperoleh grafik respon dengan *root mean square error* 0.0393063 untuk sudut azimuth dan 1.1621708 untuk sudut elevasi.
2. Nilai K_P dan K_I untuk sudut Azimuth dan sudut elevasi dapat berubah setiap waktu.
3. Respon sistem *turret-gun* dengan *Neural Network Proportional-Integral controller* dapat menurunkan %overshoot 63,493%, *settling time* 19,183%, dan *steady-state error* 27,064%. Sementara untuk sudut elevasi, *Neural Network Proportional-Integral controller* hanya dapat menurunkan *steady state error* yaitu 34,168 %. Sedangkan untuk *settling time* mengalami kenaikan 16,463% dan %overshoot juga mengalami kenaikan 75,465%.

6.2 Saran

Adapun saran yang dapat digunakan untuk pengembangan selanjutnya adalah sebagai berikut:

1. Data yang digunakan untuk pelatihan *neural network* dapat ditingkatkan sehingga hasil dari *neural network* lebih akurat
2. Perlunya penambahan *fuzzy inference sistem* yang dapat meningkatkan pengambilan keputusan pada model *neural network* sehingga hasil yang diperoleh lebih optimal.

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

- [1] Shue, S., Hargrove. C., dan Conrad, J., 2012. **Low Cost Semi-Autonomous Sentry Robot**, **IEEE Southeastcon Proceedings** 1,1:1-5
- [2] Nasyir, T., Pramujati, B., Nurhadi, H., dan Pitowarno, E., 2014. **Control Simulation of An Automatic Turret Gun Based on Force Control Method**, *IEEE Xplore Digital Library* 1,1:13-18
- [3] Pitowarno, E., 2006. **Robotika Desain, Kontrol dan Kecerdasan Buatan**. Yogyakarta: Andi Yogyakarta
- [4] Rojas, R., 1996. **Neural networks: A Sistematic Introduction**. Springer-Verlag, Berlin
- [5] Xingjia, Y., Lihai, G., Qinding, G., dan Xiaoyan, M., 2010. **RBF Neural network Based Self-Tuning PID Pitch Control Strategy For Wind Power Generation Sistem**. International Conference on Computer, Mechatronics, Control and Electronic Engineering 1,1:482-485.
- [6] Gou-Jen, W., 2001. **Neural-Network-Based Self-Tuning PI Controller for Precise Motion Control Of PMAC Motors**. IEEE Transaction on industrial Electronics, vol. 42, No.2, 1,1:408-415
- [7] Nise, N., 2011. **Control Systems Engineering Sixth Edition**, California State Polytechnic University, Pomona.
- [8] Aswam, A., Maret. 2015. **Konsep perancangan Sistem Kontrol**,
<URL:<http://www.digilib.itb.ac.id/files/disk1/611/jbptitbpp-gdl-agungaswam-30508-3-2008ta-2.pdf>>.
- [9] Gurney, K., 1997. **An Introduction to Neural networks**, University of Sheffield, London
- [10] Astrom, K. J., Murray, R. M., **Feedback Sistesms: An Introduction for Scientist and Engineering**, Princeton University Press, New Jersey
- [11] Mushonnifah, S., Subchan., Nurhadi, H. **Numerical Simulation of Acceleration and Force Control in Turret**

Gun Sistem with Disturbance, IEEE Xplore Digital Library
1,1:13-18.

LAMPIRAN

Tabel 1 Hasil simulasi sistem dengan set poin -60 sampai 60 derajat untuk sudut azimuth

Set Poin Sudut Azimuth (derajat)	K _P	K _I	Output Sudut Azimuth (derajat)	Error
-60	3	$7,06 \times 10^{-6}$	-60,09970925	0,099709246
-50	3	$7,08 \times 10^{-6}$	-50,06165005	0,061650046
-40	3	$7,08 \times 10^{-6}$	-40,0429099	0,042909903
-30	3	$7,09 \times 10^{-6}$	-30,02534124	0,025341243
-20	3	$7,10 \times 10^{-6}$	-20,01479897	0,014798973
-10	3	$7,10 \times 10^{-6}$	-10,00779133	0,007791332
0	3	$7,10 \times 10^{-6}$	0	0
10	3	$7,11 \times 10^{-6}$	10,00779141	0,007791407
20	3	$7,11 \times 10^{-6}$	20,01480048	0,01480048
30	3	$7,12 \times 10^{-6}$	30,02532019	0,025320186
40	3	$7,13 \times 10^{-6}$	40,04282197	0,042821974
50	3	$7,13 \times 10^{-6}$	50,06051082	0,060510824
60	3	$7,15 \times 10^{-6}$	60,09844633	0,098446326
Error maksimum				0,099709246

Tabel 2 Hasil simulasi sistem dengan set poin -20 sampai 60 derajat untuk sudut elevasi

Set Poin Sudut Elevasi (derajat)	K_P	K_I	Output Sudut Elevasi (derajat)	Error
-20	9,641604	0,748267	-20,2914	0,29137
-10	9,710214	0,806031	-9,64405	0,355949
0	9,605465	0,717977	-0,5891	0,589105
10	9,710836	0,806554	10,36257	0,362574
20	9,738601	0,829796	20,67934	0,679338
30	9,684369	0,78428	30,09494	0,09494
40	9,775457	0,860235	41,19184	1,191841
50	9,657024	0,761241	49,84274	0,157259
60	9,749058	0,838492	60,81172	0,811724
Error maksimum				0,503789

BIODATA PENULIS



Penulis dilahirkan di Ciamis, 11 Oktober 1993, merupakan anak ketiga dari 3 bersaudara. Penulis telah menempuh pendidikan tingkat dasar di Sekolah Dasar Negeri 2 Sandingtaman. Sementara untuk pendidikan menengah, penulis menimba ilmu di SMP Negeri 1 Lumbung dan SMA Negeri 2 Tasikmalaya. Kemudian setelah mengikuti Seleksi Nasional Mahasiswa Perguruan Tinggi Negeri, penulis diterima di Jurusan Teknik Mesin FTI-ITS pada tahun 2012 dan terdaftar dengan NRP 2112100112.

Di Jurusan Teknik Mesin ini, penulis mengambil Bidang Studi Manufaktur. Penulis sempat aktif di beberapa kegiatan Seminar yang diadakan oleh Jurusan dan aktif di organisasi tingkat jurusan yaitu Dimensi Mesin ITS. Selain itu, penulis juga aktif sebagai Asisten dan Grader Praktikum Pengukuran Teknik.